Chapter 1

INTRODUCTION

1.1 Introduction

The need for information anywhere and at any time has been the driving force for the increasing growth in mobile networks and devices. The field of mobile computing is the merger of advances in computing and communications with the aim of providing seamless and ubiquitous computing environment for mobile users. Whereas notebook computers and personal digital assistants (PDAs) are self-contained, networked computing constitutes a new paradigm of computing that is revolutionizing the way computers are used. Mobile networking greatly enhances the utility of carrying a computing device. It provides mobile users with versatile communication to other people and expedient notification of important events, yet with much more flexibility than cellular telephones and pagers. It also permits continuous access to services and resources of the traditional land-based wired networks. This combination of networking and mobility will engender new applications and services, such as collaborative software to support impromptu meetings, electronic bulletin boards that adapt to the contents according to the participants present, self adjusting lighting and heating, and navigation software to guide users in unfamiliar places and tours. To support mobility in the Internet, the Internet Protocol (IP) has been extended to support mobility. Also at the same time, there is also a growing trend for these IP based networks to operate in an infrastructureless environment called mobile ad-hoc networks. However, the proliferation of such mobile networks depends on a multitude of factors, with trustworthiness being one of the primary challenges to be met. Despite the existence of well-known security mechanisms, additional vulnerabilities and features pertinent to this networking paradigm might render traditional solutions inapplicable. One of the main difficulties in promoting the concept of mobile computing is that, inherently these networks are extremely vulnerable to security attacks as they not only rely exclusively on unreliable and insecure wireless communication but also the nodes in the network usually may move from one administrative domain to another. This is further complicated by the fact that the mobile networks are typically characterised by severe constraints in resources such as bandwidth and battery power making implementation of security all the more challenging. The objective of this dissertation is to address the issues involved in the design of security services for Mobile IP and ad-hoc networks. This introductory chapter presents the overall layout of this thesis and defines its objectives.

1.2 Thesis Structure

1.2.1 Scope

Extensions to IP based networks (both wired and infrastructureless networks) to facilitate mobility have not been designed keeping security in mind. However adequate security features are basic requirements for the continued functioning of mobile networks. Clearly the problem is so broad that there is no way to devise a general solution. The complexity and diversity of this field has led to a multitude of proposals which focus on different parts of the problem domain. We aim to address most of these wide-ranging problems and in the process initiate a practical approach.
to the development of an integrated security infrastructure for mobile networks. The intention is to seamlessly integrate these security services and mechanisms at the IP level within the mobile IP and ad-hoc networks. The provision of security services at the higher and lower layers and their interoperability with our proposed framework is outside the scope of this thesis.

1.2.2 Organisation

The thesis is divided into four parts: A, B, C, and D.

Part A offers an introduction to the field of Mobile Computing. This section includes chapter 1. This chapter is an introductory chapter on mobile communications. It introduces the concept of mobile computing and communication and then highlights the applications of mobile computing and thereafter describes some technical design challenges in realizing a mobile computing environment.

Part B deals with location management and security issues in Mobile IP based networks. It addresses the general problem of mobility management and packet routing to mobile hosts. This Section includes chapters, 3, 4, and 5. Chapter 3 introduces the concept of location management in IP based mobile networks. It then presents an overview of location management schemes with an emphasis on basic approaches used in the Internet. It then highlights the significance of security for these networks and finally concludes with a survey on popular proposals that aim to provide security for Mobile IP based networks. In chapter 4, we introduce our architecture for location management known as the distributed location management (DLM) model in mobile IP based network with rationale and design principles. We then compare our architecture with other major proposals that were introduced in the previous chapter. chapter 5 proposes a security framework for the DLM model.

Part C deals with Secure IP Multicasting in mobile IP based networks. It includes chapters. In chapter 6, we discuss issues in carrying multicasting for mobile IP based networks. It also provides a background on proposals that aim to facilitate secure multicast service in mobile IP based networks. It then describes our solutions to providing security for multicast service in Columbia and IETF based mobile IP networks. Our framework caters to both key management and anonymity services.

Part D deals with Ad-hoc Networks. It addresses the problems related to security, location management, and Quality of Service in ad-hoc networks. It includes chapters 7, 8, 9, 10, and 11. In chapter 7, we introduce the concept of infrastructureless networks called mobile ad-hoc networks. In chapter 8, we provide a comprehensive survey of proposals that that aim to provide security for mobile ad-hoc networks. In chapter 9, we provide a security framework for near term digital radio (NTDR) ad-hoc networks, which, caters to end-to-end secure communication, key management and anonymity. In chapter 10, we extend our security framework to facilitate secure routing services for ad-hoc network. In chapter 11, we describe the design principles of security layer for ad-hoc networks. In addition to providing security, this layer also caters to basic quality of service (QoS) features. We provide differentiated service
based QoS architecture for NTDR ad-hoc networks. This architecture is applicable to any cluster based ad-hoc network environment.

Finally research conclusions and recommendation for future work are presented in chapter 12.
Part A
Mobile Computing
Chapter 2

Introduction to Mobile Computing

2.1 Introduction

Mobile Computing has become tremendously popular in recent years. The key factors that have contributed to its growth are miniaturization of mobile computing devices and availability of more processing power in these devices. This has in turn initiated the development of better computing based applications for these devices. On the other hand, the market for wireless telephones and communication devices is experiencing a rapid growth [Perkins 2001]. The traditional LANs have also been extended with wireless interfaces portable computers capable of wireless access. In addition, new satellite services have been proposed whose initial applications are predominantly voice and paging. This rapidly expanding technology of cellular communications, wireless LANs, and satellite devices will make it possible for mobile users to access information anywhere and at any time [Imielinsky+ 94]. As a consequence, the resulting computing environments often called mobile or nomadic computing, no longer restricts a user to maintain a fixed location in the network. Users can now be mobile and will be able to send and receive information from any location.

One must be able to distinguish between the terms mobile computing and wireless networking. These two terms are not synonymous. Computing on the move is possible with a wired connection to the network. However, the presence of an underlying wireless infrastructure provides greater flexibility of movement in a seamless way. One must also be able to distinguish between two similar terms: Mobility and Portability. In the portable operation environment, the device can be operated at any of a set of points of attachments, but not during the time that the computer changes its point of attachment. In other words, if the device is moved from one place to another, then its network connections have to be shut down and reinitialised at the new point of attachment to the network. Whereas in a truly mobile operation, a device can remain in almost continuous contact with the network resources needed by its applications. In other words, neither the system nor any of the applications on the system need to be reinitialised or restarted, even when network connectivity is frequently broken and re-established at new points of attachment.

While the integration of wireless infrastructure with mobile computing offers several advantages it also gives birth to challenges especially when one considers porting network application over such architecture. We begin this chapter first defining and identifying uses of mobile computing networks in section 2.2. In section 2.3, we classify the different types of mobile networks. In section 2.4, we shall explore the technical challenges in designing and building mobile computing systems. Finally in section 2.5, we present our concluding remarks.

2.2 Mobile Computing: Definition and Uses

Mobile Computing refers to the use of Mobile Devices to access information through wired or wireless connection anywhere. Mobile devices are no longer simple devices.
Mobile Computing can be seen as an integration of portable computers and wireless networks. It can also be seen as a combination of portable computers, modems and telephone network [Nguyen+ 95]. Nowadays, most mobile devices are equipped with:

- Colour display (up to 65,535 colours),
- Chinese language support (input and display),
- Larger memory (up to 32 MB built-in memory plus removable memory card),
- Faster wireless communication interfaces (up to 14.4Kbps GSM modem connection),
- Better security support (Web Browser with Secure Sockets Layer (SSL) support)

Furthermore, with the support of application development tools and communication devices, some mobile devices can be used to get access to office information systems. Technological advancements have and will continue to improve battery performance and increase wireless-network bandwidth, but the resource scarcity is relative. Mobile computing environments will continue to be resource-poor compared to their fixed counterparts, at least in the foreseeable future.

Some typical applications of mobile computing are described below.

1. **Mobile Form Filling - Meter Reading, Site Inspection and Interview**

Some users may be required to work outside their offices to collect information (such as taking meter readings, site inspection statistics and interview result). These users can use mobile device such as a Palm-size PC to collect information at the remote sites. The collected data stored in the device will then be transferred back to the server remotely through a mobile phone or locally through a desktop PC after the users are back to the office.

Since data are input directly to the mobile device and transferred back to the server afterwards, there is no need to re-input the data. This practice eliminates transcription errors and thus improves both data accuracy and user productivity. However, due to the small screen size of mobile device, only simple forms can be used in these applications.

2. **Mobile Access to Office System - Inventory System and Inquiry System**

Some users may not have a fixed work place while some others may need to work outside the office premises. In these cases, the mobile user can use a mobile device such as a laptop to access the office systems such as Mainframe, Unix and Windows NT applications through a terminal emulation or client software (3270 emulator, Telnet client and Windows Terminal Server Client respectively) anywhere beyond the office boundary.

This solution is also suitable for indoor workers when the work place environment is not convenient or suitable to install wired connection.
3. Mobile Email

Email is currently one of the most popular office applications. Mobile users can use a laptop or any other device to connect to the mail server to retrieve emails from or send emails to the corporate email server through a mobile phone.

The user can download email from the server and store them in the laptop or Palm-size PC and read the email off-line. Also, the user can prepare a new email or reply to mail off-line with the mobile device and send back to the server when it is connected to the server. However, since certain mobile devices, such as Palm-size PC, are only equipped with a small amount of memory, the size of email to be downloaded should be controlled by setting a limit. Moreover, Palm-size PC may not have the corresponding application or viewer to process email attachments.

4. Mobile Internet Access

A typical application area of mobile devices is its deployment in election affairs. During the period of an election, statistical data can be announced through a Web Site in the Internet. The concerned personnel can view up-to-date statistical data through a Web browser running on Palm-size PC or Handheld PC.

Users can use a Handheld PC or Palm-size PC to browse the Internet to obtain real-time and up-to-date statistical information. Since the Web browser on a Handheld PC or Palm-size PC is not as powerful as those installed in a desktop or notebook PC, some features like frame-set, Java and some other add-ons are not supported. Moreover, a simplified web page may be required for Handheld PC or Palm-size PC to match the small screen size and slow communication speed of wireless connection.

5. Short Message Services - Mobile Field Services Management

In an order dispatch centre, operators can send orders and other related information to the mobile phones of their Field Services Staff using Short Message Services (SMS). On the other side, the Field Services Staff can send another SMS messages back to the dispatch centre to acknowledge or report the order status.

SMS messaging is more accurate and informative than a verbal order, and so it can improve efficiency and minimise operation cost. As the SMS message has a limited size and the screen of a mobile phone is quite small, the service application should be kept simple.

2.3 Classifying Mobile networks

Mobile networks can be broadly classified as follows:

**Mobile Voice Networks:** A cellular-like network with a wired infrastructure and wireless connection between a user terminal and the network infrastructure. A cellular network can be analogue or digital.

**Analogue Cellular Network:** Analogue cellular networks, sometimes referred to as Analogue Mobile Phone Systems (AMPS) are based on a short-wave analogue
transmission via a wireless connection from a mobile telephone (e.g. mobile station or handset) to a transmitter station (which typically relays the call on to the conventional public telephone network).

Analogue cellular networks are based on transmission regions known as cells (hence cellular network). The size of a cell is determined by a combination of the transmitter or base station’s transmission power and the local topology. Cellular telephone networks are normally limited to urban areas, and along major road, rail and river systems where there is sufficient subscriber density to provide a business case for location of a base transmitter. As the cellular telephone user moves from one cell or area of coverage to another, the telephone call signal is passed on from one to the next local cell transmitter in a process is known as hand-over. Cells are typically arranged as hexagonal structures, and the structure of cellular networks can be visualised as an overall honeycomb, or tessellated hexagonal structure.

The Analogue cellular network has a number of disadvantages. It provides limited bandwidth from phone to transmitter, and uses the available radio frequency space less efficiently than contemporary digital mobile phone systems. These networks are also more prone to noise interference as there are fewer opportunities for error correction in analogue systems than in digital systems. The combination of inefficient bandwidth usage and susceptibility to interference and noise results in a poorer quality of service for end users than is available through digital cellular networks. In these networks, the radio signals are not encrypted and can be monitored by third parties utilising very low cost radio scanners. Due to these disadvantages, Analogue cellular technology is commercially threatened and technologically superseded by digital technologies such as CDMA/TDMA, GSM and DECT. Analogue cellular technology is commercially threatened and technologically superseded by digital technologies such as CDMA/TDMA, GSM and DECT.

**Digital Cellular Networks:** There are several standards under the umbrella of digital cellular networks. Some key standards that we shall describe in this section are as follows:

- **Group System Mobile (GSM) Networks.**
- **Code Division Multiple Access (CDMA) Networks.**
- **Time Division Multiple Access (TDMA) Networks**
- **Personal Communication Networks (PCS) Networks.**

**GSM Networks:** Global system for mobile communication (GSM) is the most widely used form of digital wireless technology for mobile telephony and data communications in Europe. (The meaning of the GSM acronym used to be Group Système Mobile). GSM hand held units (typically mobile phones) are uniquely identified by their Internal Mobile Equipment Identity number. The subscriber is identified by an International Mobile Subscriber Number; these numbers are theoretically independent of the GSM network. The handheld unit communicates with a local Base Transceiver Station (BTS). A base transceiver station manages radio communications links with the hand held set. In large urban areas there may be a number of BTS systems. The BTS systems are connected to a Base Station Controller (BSC), which manages radio channel set-up and hand-over between
BTS's facilitating roaming communication. The BSC acts as a through connection point between mobile hand held units and mobile service switching centre (MSC).

A mobile service-switching centre acts the same as a switching node on a PSTN or ISDN cable network. A mobile switching centre is responsible for connecting calls together by switching the digital voice data packets from one network path to another - a process usually called 'call routing'. In addition MSC's provide additional information to support mobile service subscribers, including user registration, authentication and location updating. MSC's are also the point of fixed connections through to PSTN and ISDN networks.

The GSM technology provides many benefits. Since it provides a large-scale open telecommunication standard for manufacturers, it creates a large market for interoperable hand held mobile stations and other devices. It also provides a flexible technological architecture and support for intelligent networking concepts. GSM has some disadvantages. However, most problems with GSM systems are not technical but service related. Until GSM devices become common, current marketing and operational practices restrict the trans-European functionality of the system by effectively levelling an operator level tariff for trans-European or Out of Country Calling. Such practices limit the adoption of GSM as a single mobile communications solution. A technical related problem is that of communications caused by interference between the cell-phone and base station. The sole solution to this is more base stations. It is facing increased competition from DECT systems that are cheaper (as it is a radio access technology rather than a comprehensive system architecture); more secure as security features based on cryptographic techniques have been included in a systematic way for the first time [Chen+96], and do not require operational licenses.

- **Code Division Multiple Access** (CDMA) is a digital wireless transmission technology for mobile communications. CDMA is one of a number of approaches to implementing widely distributed digital mobile voice and data networks. It is viewed as a competing technology to European de facto mobile digital wireless technology GSM.

As with the majority of wireless transmission and communication network technologies, CDMA uses a hand held transceiver, which transmits calls to a localised base station. The CDMA standard defines 64 channels transmitted from the base station and can support a maximum of 63 simultaneous users per 1.25-MHz frequency. Designed as an improvement on analogue cellular and Time Division Multiplexing Access (TDMA) telephony, CDMA is being promoted by some industry actors as the next mobile data networking technology.

CDMA uses a technology called direct sequence spread spectrum transmission. This is a form of multiplexing where the transmitter (the mobile phone) encodes the signal using a pseudo-random code sequence that the receiver (base station) also knows. The receiver uses this code to decode the received signal. Each different random sequence corresponds to a different communication channel. CDMA is the equivalent of 2nd generation GSM in Europe, and the emerging CDMA1 is equivalent to the emerging 3rd generation GSM.
Differing in technology to GSM, CDMA is more efficient and provides greater capacity than GSM, TDMA and analogue cellular telephony. However, GSM is designed to support data as well as voice transmissions. Current CDMA network implementations do not offer data communications services.

- **Time Division Multiplexing Access** (TDMA) is a mobile communications system based on digital technology that operates in the same radio frequency range as traditional analogue cellular systems. TDMA is an alternative approach to CDMA and GSM technologies for the development of widely distributed digital mobile voice and data networks other than GSM. TDMA is a cellular telephone network technology. A TDMA telephone set communicates with a base station radio transceiver. The area supported by a base station is called a cell (hence cellular).

  Like GSM, TDMA digitises voice signals and subsequently compresses the data, before sending the data to a local base station. As with other cellular systems such as GSM, TDMA manages the hand-over between cells so the user can have free roaming communications. Time division multiplexing access, as its name implies, employs two channels of user information, transmitted over the same link by allocating a different time interval for the transmission of each channel i.e., the channels "take turns" to use the link. Each channel has a bandwidth capacity of 24.3 Kb/s. Current implementations digitise speech at 8 Kb/s which enables three times as many phone conversations to be hosted along the same communications channel as traditional analogue cellular systems on the same frequency.

  TDMA offers a business advantage, as it is already an established digital mobile telephony standard in North America and Canada, which utilises bandwidth more effectively than comparable systems. However, it has a number of disadvantages. For instance it has poor voice quality and has limited service implementation over existing cellular systems. It also does not provide support new digital services such as Caller ID etc. Currently it is facing a strong competition from CDMA and GSM technologies.

- **Personal Communication Networks** (PCN) is a broad term used to describe a number of developments in mobile digital telephony. To some it is the development of integrated services to the point where paging, alphanumeric messaging, voice and data services can be delivered to a user in one device. To others it is a new mobile digital communications infrastructure, which provides the facilities to support concepts such as personal numbering, service selection and unified billing.

  At its essence, personal communications networks are about the development of new mobile digital communications infrastructure which can deliver personalised people centric services in a convenient manner to the end user, in any environment, be it fixed, domestic, commercial, or mobile free roaming.

  The concept of personal communications networks (PCN) was officially launched by the Department of Trade and Industry (DTI) in the United Kingdom in 1991 and subsequently adopted by the European Telecommunications Standards Institute (ETSI), in order to prevent the proliferation of proprietary standards.
throughout Europe. Personal communication networks are centred around the twin concepts of deploying extensive digital cellular networks and the notion of a unique identification number called a Universal Personal Telecommunication (UPT) number.

PCN digital cellular networks differ in implementations to existing cellular communications systems such as DECT and GSM. Standard cellular telephony is based upon the concept of a mobile handset, which transmits to a local radio transceiver. This radio transceiver is called a base station and the area it covers is referred to as a cell. The size of cell is dependent upon the power of the base station transceiver and standard cellular telephony cells have diameters of several kilometres. PCN employ micro-cells, which have diameters of several hundred metres and are spaced no more than 650 metres apart. In order to cover the same area as a regular cellular telephony cell, a number of micro-cells are used. They permit individual buildings to establish restricted personal communications networks which can be used for private wireless voice and data networks and fulfil increased capacity requirements in areas which possess high user density. They also permit lower transmission power (0.1 - 1.0 Watt), which enables smaller, lighter portable phones with longer talk times to be used.

At the core of the PCN concept is the idea that each subscriber is assigned a personal identification number. This number identifies the subscriber to the network and enables them to receive or initiate phone calls, regardless of their respective location. All universal personal telecommunication numbers are held in a networked database. As a PCN mobile phone moves from one micro-cell to another it uses the signalling network to notify the network that its location has changed. Alternately as a call for a given number enters a switching exchange, the exchange triggers a signalling system request [see Signalling systems] across the network to look up in the database how to handle the call. The database enquiry returns information on how the call should be handled. All local micro-cell controllers in an area are polled to find out if the called party's phone is switched "on", and if so where it is. Each local micro-cells issues a page across the networks electronic paging channels, much the same as existing GSM cellular networks do, and in response a paging signal is generated by the user's phone to the nearest area controller, to say 'here I am'. The call is then switched through the network to the phone.

A communication in a Personal Communication Network can take the form of voice, paging or data communications; the network is designed to accommodate all forms of digital communication. Products for PCN increasingly have voice, personal paging and electronic messaging services capabilities built in. PCN aims to provide a unifying communications service package to the end user, including: roam anywhere wireless phone services, answering machine/voice-mail, message or numeric paging, caller ID, conference and call forwarding to end users.

PCN offers the advantage to the end user of a flexible, personalised communications service capable of integrating voice, paging and messaging services within one device and one bill. However, in order to deliver the promise of personal communications networks and their related services, substantial investments are required to implement the underlying micro-cellular
infrastructure. Currently although GSM cellular networks are widespread, the
development of European PCN infrastructure has been limited to small DCS 1800
networks being developed in parts of Germany and the UK.

Observations

Cellular Networks primarily provide voice and limited data services to users with
hand held phones. Their coverage only extends to metropolitan areas, and they have
problems with scalability (to accommodate a large number of mobile users), data
transmission, and low bandwidth for data intensive applications.

Nomadic Networks: This is a case where nomadic users with wired network
connectivity have access to the network at attachment points. A nomadic user might
disconnect from an access point before a move and might later reconnect with an
access point. In the interim period, the user operates in disconnected mode, relying
solely on information resident on the mobile computer. In other words, a seamless
communication is not possible here. Using nomadic computing technology, people
will be able to easily access services, content, and people while they are on the move
and at arbitrary destinations. The protocol that tries to achieve such computing by
providing mobility at the network level is mobile IP. Mobile IP [RFC 2002] was
suggested as a means to attain wireless networking. It focuses its attention at the
Network Layer, working with the current version of the Internet Protocol (IP version
4) [RFC 791]. In this protocol, the IP address of the mobile machine does not change
when it moves from a home network to a foreign network. In order to maintain
connections between the mobile node and the rest of the network, a forwarding
routine is implemented.

Microwave Networks: Microwave networks are a wide area communications system
that uses the microwave end of the electromagnetic wave spectrum as a transmission
medium. Microwave transmissions take place in the 3 to 30 GHz range of the
electromagnetic spectrum. Terrestrial Microwave Networks operate over distances of
up to 30 miles, between pairs of transmitting and receiving antennas, although the
higher the frequency of transmission, the shorter the distance across which
transmission can be made. In order for microwave antennas to achieve clear
transmission they must be an uninterrupted line-of-sight between pairs. At the
midpoint between the transmitter and receiver, the beam can spread in diameter up to
a few dozen metres and this area, known as the Fresnel Zone, must be completely
clear of obstruction such as trees, buildings, or hills. The maximum distance between
microwave transmitters is dictated principally by the curvature of the earth, although
atmospheric conditions tend to mean that distances less than the theoretical limit are
chosen for transmitter receiver pairs for the practical purpose of guaranteeing
throughput in the majority of weathers. Water, in the form of humid air, fog or rain
absorbs microwave energy and can disrupt transmissions. Microwave
communications over long distances are achieved by the use of series of microwave
repeaters, which receive, re-amplify and retransmit a signal to the next post.

Microwave is the basis of satellite communications; earth stations transmit data to
satellites, which receive, re-amplify and re-transmit the information to a
geoographically remote earth station. Microwave networks are also used to provide a
high bandwidth network infrastructure for broadcast transmission (television) and
long distance two-way communication (voice and data). Increasingly digital microwave transmissions are being used to provide a digital bypass for congested cable networks in order to cope with massive bandwidth requirements.

Microwave networks offer a number of benefits. For instance these networks are a very convenient way of moving relatively high bandwidth data across short distances, and are less costly to install than cable systems. One of the most common current applications is to provide data links between mobile phone cells (the transmitter/receiver at the heart of a geographical call) and telecommunications switching centres. They are also capable of supporting data intensive applications such as real-time video transmission. Microwave networks are probably most commonly used for television broadcasts between regional relay transmitters, where the signals are then re-broadcast on longer wavelength radio frequencies. Microwave networks can provide a digital bypass for the optical fibre and cable based networks. There are some disadvantages with its usage too. The greatest limitation of microwave networks is that it requires a “line of sight” transmission restricting their application in urban areas. Microwave transmission is also susceptible to atmospheric interference including rain absorption (the rain drops are heated up as they absorb microwave radiation). It has also been experimentally proved that microwave transmissions are hazardous to health and care is needed to ensure that humans and animals do not intercept a microwave beam near its source.

There is another technology called Wireless Wide Area Networks (Wireless WANs). This is a special type of mobile radio network that provides wireless email services or wireless access from a mobile host to an application running on a fixed host. An example of such a network is Ardis, a two-way store-and-forward, packet-based wireless network. The coverage of a wireless WAN extends for a wide area, but it has low bandwidth for data services and may have a problem with scalability.

**Public Mobile Radio:** Public mobile radio is a two-way radio system that functions as a wireless multi-party communication system for short distances. It permits two or more radio operators to communicate over a range of several kilometres. The most popular form of public mobile radio is commonly known as Citizen Band Radio. Public mobile radio sets are commonly found in cars, lorries, and other vehicles so that the driver can obtain road and weather condition information from operators in nearby vehicles or call for help in an emergency. Probably the most common form of public mobile radio encountered is used by taxi drivers to keep in touch with their control (booking centre). Wavelengths and frequencies, as well as licensing requirements, vary from country to country.

Public mobile radio can provide a simple low cost immediate communications link that can cover a range of several kilometres. However it not a secure or private transmission medium and as a result is only used for limited business communications.

**Paging Networks:** Paging systems provide a mobile alphanumeric message communication service. Messages are displayed received and displayed on a small handheld device, called a pager. Paging systems are typically used to convey contact information which to people who may be unable to answer a telephone.
A pager is a small telecommunications device that receives and displays short messages. Pagers are the 'mobile stations' of paging systems. They are generally used by people who are continually changing their location and may not necessarily be able to answer a phone call immediately (e.g. a doctor or a midwife). Typically, a pager captures a message (usually accompanied by a beep or vibration to alert the user), which consists of very short text messages or a phone number to ring. Paging systems function in similar ways to mobile phone cellular networks and messages are often sent using local digital mobile networks personal messaging services.

Paging systems are relatively cheap to run, small and unobtrusive to use (unlike mobile phones) and provide limited text and numerical communication. Paging systems have, historically, been unidirectional, capable only of receiving messages. Rival developments such as GSM based telephones, provide support for the short messaging service used by paging systems and can display messages of 165 alphanumeric characters.

**Wireless Local Loop (WLL):** WLL is a telecommunications network that provides a telephony service using radio as a substitute for the more traditional cable based local loop. WLL, sometimes called FRA (Fixed Radio Access) or RITL (Radio in the Loop), connects users to the PSTN (Public Switched Telephone Network). The connection uses radio signals, as opposed to copper wire, for part or all of the connection between the switch and the user ('last mile' connection). The connection is provided by proprietary fixed radio access, fixed cellular systems, or cordless access systems.

There are a number of advantages associated with WLL. WLL telecommunications transmission is more effective, and is more easily deployed, than fixed services in rural areas and areas with adverse terrain. WLL can also be deployed very quickly, with providers able to deploy a system in between 90 and 120 days. The expense of providing services via WLL is not affected by the distance between the user and the provider, making WLL less costly than fixed line services. WLL bypasses existing wire-line networks, therefore, competition will open up, and prices should be driven down. Telecommunications companies see WLL as a major area for expansion in rural areas and in local subscriber growth. This is due to costs preventing implementation of fibre/wire networks. WLL has a far lower initial cost than copper wire, and is cheaper to distribute at low subscriber densities. The cost of deploying wireless connectivity is expected to fall, whilst copper deployment costs will remain constant. However there are some issues that need to be addressed. There are no specific WLL standards, potential providers are faced with a perplexing choice between mobile, fixed-access, and digital cordless applications. The appropriate technology can only be decided upon when a sizeable level of consideration has been given to size and population density of the area, and service needs of the user base. Being a relatively new technology, WLL cost models are sometimes inaccurate. Some ancillary expenses in the distribution and maintenance of the system are not accounted for.

**Wireless Local Area Networks (Wireless LANs):** Wireless LAN technology can be cited as an example of realising nomadic computing within a limited area. Wireless LAN related solutions focus on providing mobility at the link layer. Wireless LAN facility can be connected to a mobile computer or a fixed network via a wireless
interface card that has an antenna. The major motivation for and benefit from wireless LANs is increased mobility. Untethered from conventional network connections, network users can move about almost without restriction and access LANs from nearly anywhere. The wireless LAN resembles a cell in a cellular network.

IEEE's proposed standard for wireless LANs is titled IEEE 802.11 [Chen 94]. It runs in two modes: Infrastructure and ad-hoc. In the infrastructure mode, the architecture uses fixed network access points with which mobile nodes can communicate. These network access points are sometime connected to landlines to widen the LAN's capability by bridging wireless nodes to other wired nodes. If service areas overlap, handoffs can occur. This structure is very similar to the present day cellular networks around the world. In the ad-hoc network, computers are brought together to form a network "on the fly." We discuss ad-hoc networks in the next section. The problem with wireless LANs is that they only provide coverage for a local area, e.g., inside a building, and do not provide networking support for wide area moves.

There are several issues that need to be addressed before a nomadic network can be realised. The access must be “plug-and-play,” that is, transparent to the nomad or the moving device and its applications. This implies that there should be no loading of special software, and also no change with laptop or network configurations. In other words a way is needed to automatically and transparently adapt all aspects of the user’s computing, communication, and storage functions to any device in any environment. Also additional nomadic services are required to manage the network. For example, there must be provisions to locate network resources or to help a device adapt to changing link conditions and so on. This is a tall order, but the technology is evolving to make this happen.

Mobile Ad-hoc Networks: A Mobile Ad-hoc Network (MANET) with no wired infrastructure. In this model, all the nodes are mobile, and communication is over wireless links. Because paths between nodes might comprise multiple wireless links, each node should be capable of acting as both a communication node as well as a router with packet forwarding functions. A MANET might have a gateway that connects it to other networks. In the simplest form, all MANET nodes are identical. Other possibilities could be, for example, a hierarchical mobile network where some nodes are more powerful, have an abundant energy supply, and collectively support the relatively resource-poor nodes.

There are a number of issues that need to be addressed before an ad-hoc network can be realised. In particular, routing issues in a dynamically changing topology, communication between heterogenous devices, security and quality of service related issues coupled with regular problems of mobile networks pose a formidable challenge that needs to be urgently addressed.

2.4 Design Issues in Mobile Computing

In this section we will discuss some of the key issues involved in realising a mobile wireless computing environment.

1. Data Link Layer Issues: With the increasing popularity of mobile computing and communication systems, the demand for a limited number of wireless
channels is increasing. The key issue is to utilize the channel efficiently. Most non-real time packet based applications do not require a dedicated channel. However, Quality of Service (QoS) based real time applications must be guaranteed some dedicated bandwidth. Some prominent reservation based issues are:

- Channel allocation for cellular networks
- Contention based approaches for wireless LANs.

**Channel allocation:** Channel allocation schemes are appropriate when the connections between mobile host and the base station are long lived. Proposed techniques take one of the two approaches: Centralised or distributed. Centralized solutions are simpler and always result in channel utilization that is at least as high as that for distributed solutions; however, such a solution has a single point of failure and does not scale well.

The other related issue is whether the allocation technique is fixed or dynamic. In a fixed channel allocation (FCA) scheme a fixed number of channels is assigned to each cell in the cellular network. It can result in poor channel utilization, especially if the spatial distribution of demand is non-uniform. Distributed channel allocation schemes result in better channel utilization but tend to be more complex and require more control messages between base stations.

A third alternative is hybrid channel allocation, which divides the set of channels into a fixed and dynamic set. This solution divides channels in the fixed set into subsets of nominal channels associated with each cell and allocates these channels using FCA algorithms. Channels in the dynamic set are available to all cells and are allocated using distributed channel algorithms when a cell’s nominal channels are all in use.

**Contention-based approaches:** Traditional local area networks rely on carrier sense multiple access with collision detection (CSMA/CD) protocols to gain access to the transmission medium. However, in Wireless LANs, an interferer can be in the intended receiver’s communication range but the transmitter does not sense the interferer’s transmission. In this situation, the transmitter will not sense a collision even though the receiver experiences interference, and CSMA/CD will not yield desirable performance. Therefore, protocols such as IEEE 802.11 employ collision avoidance. Wireless LANs can also operate in contention free mode when the base station regulates access to the channel. So, unlike Ethernet, where control is distributed, protocols such as IEEE 802.11 support both distributed and centralized operation.

2. **Issues Relating to Location Management and Mobile IP:** Mobility is behaviour with implications for both fixed as well as the wireless networks [Imielinski+94]. On the fixed network platform, mobile users connect to different data ports at different locations. On the other hand, the presence of a wireless infrastructure enables an unrestricted mobility and connectivity from any location within the radio coverage. Mobility is an issue at the network level of the protocol stack and results in constantly changing topology of the system, calling for mobility of resources. The area of location management deals with the mobility of the clients.
as the location of a user can be regarded as a data item whose value changes with every move of the mobile host. The following design issues must be taken into consideration when considering location management of mobile hosts:

- A change in mobile node’s location implies a change in the route to that node. So a fundamental trade-off here is between searching and informing. Should the mobile host be responsible for informing all of its correspondent hosts about its change of location?
- Alternatively, should the correspondent nodes be responsible for tracking the mobile host? What if a correspondent host is incapable of supporting mobility? Does this mean that the host cannot communicate with other mobile hosts?
- Another possible option is to have some trusted entity maintaining this location information on behalf of the mobile host.
- A mobile host cannot predict which nodes might want to communicate with it. So, in cases where the mobile host is responsible for location updates, should it send these updates and the resultant route updates to the entire network or to a selected set of hosts?
- A mobile host might want to communicate with other nodes without revealing its location. In such situations, the real identity of the mobile host is not revealed to the administrative authority at the visiting network. Hence one cannot monitor the movements of the mobile host. Location anonymity might be a useful feature in a military or a battlefield environment or in such situations as investigators tracking down criminals. The other side of the coin is that a visiting network may not be willing to offer its services to a mobile host having an undisclosed identity for a wide variety of reasons such as security and billing. Hence, some mechanism must be in place that can vouch for the credentials of a mobile host without actually revealing its true identity. The key question then is how does one support such a facility?

Location management in cellular telephony employs home location registers and visitor location registers. When a mobile phone is in a foreign domain, it registers with a database called the Visiting Location Register (VLR) serving the foreign region. The VLR conveys this registration information to the mobile phone’s home location register (HLR). Subsequently, when a call for the mobile phone is routed to its home domain, the HLR forwards the call to the VLR. The VLR determines which cell the phone is in and completes the call.

Mobile IP, the protocol to deliver IP packets to mobile hosts, has a similar approach. Home agents and foreign agents correspond to the HLRs and VLRs. Unlike mobile telephony networks, mobile IP does not support real time handoff on an ongoing call between cells. For further details refer to [Perkins 98]

3. **Routing Issues**: The routing solution depends on how the underlying mobile infrastructure is organised: Nomadic, cellular, or Mobile Ad-hoc Networks (MANET)

- **Nomadic computing**: In the Nomadic computing model, it is assumed that there is no network connectivity during a move. In principle, such a facility provides portability rather than true mobility. The mobile host must connect to a network, register with the appropriate administrative authority for the network and then access services offered by that network. Irrespective of
where the mobile host is located, the packets destined for this host must be routed to this host. Mobile IP [Perkins 98] is one scheme that facilitates such communication. All the communicating nodes could route packets to the mobile node's home address. There, the home agent would intercept the packets and tunnel them to the mobile host.

**Cellular computing:** A cellular network environment offers continuous connectivity i.e., the communication remains active while the mobile device is on the move. This is accomplished by means of a handoff process. As a mobile node moves out of one cell into a neighbouring cell (or from one wireless LAN to a neighbouring wireless LAN), seamless handoff between the cells is accomplished via base stations. The key issues that must be considered are as follows:

a. Unlike handoff of voice calls in cellular networks, occasional loss of a few data packets can severely degrade the performance of several data communication applications.
b. Applications that rely on data communication in a cellular setting will become prevalent when third-generation packet radio networks become a reality. The cellular networking domain currently does not have many such applications, owing to the low available bandwidth.
c. Depending on the nature of the application, data packets might or might not have delivery deadlines and reliability requirements. So, QoS issues are important in routing decisions. The added dimension of mobility makes the problem more challenging. QoS research and solutions related to backbone networks do not necessarily address the issue of source and destination mobility.

Inter-system mobility must distribute handoff management so that calls are routed with optimum efficiency and minimum delay such that independent sub-networks can maintain service to their subscribers [Garg+ 96, Lin+ 96]

**MANETs:** Such networks are devoid of any fixed infrastructure and therefore all participating nodes must collaborate together to routing and packet forwarding functions. There are several solutions that are proposed and we examine these proposals in later chapter in the thesis. Such networks have interesting potential civilian and military applications. Some issues relating to routing in MANET are as follows:

a. In terms of volumes, how much control routing traffic needs to be propagated to maintain the network in a consistent state?
b. How long does it take for the routing algorithm to converge after a change in topology?
c. What are the potential failure points in the routing algorithm that might leave the network in an inconsistent state?
d. Does the protocol determine routes that are optimal with respect to the metric that is being optimised? Some protocols try to minimize the number of links on the path between the source and destination. Others might have different goals, such as load balancing among all nodes, minimizing energy consumption, or minimizing latency.
e. How stable are the routes?
f. Wireless links might be unidirectional. Can the routing protocol utilize such links? If so, does the data link layer need to be modified?
g. What fraction of packets is dropped owing to non-availability of routes even though the source and destination are part of the same connected component?
h. What fraction of packets is misrouted?
i. What is the level of fault tolerance supported by the routing algorithm?

There are a plethora of proposals for accomplishing routing in ad-hoc networks. These proposals are discussed in detail in a later chapter.

4. Transport Layer Issues: At the transport layer, only end-to-end session issues matter. However, the underlying data link layer in a mobile network is susceptible to high bit error rate and low bandwidth that can significantly affect the performance of Transmission Control Protocol (TCP) [RFC 793] connections. Consider a TCP connection in a cellular system where only one link (the last or first) is wireless. Packets dropped across a wireless link will result in TCP time-outs and retransmissions by the source. Retransmitted packets will travel over the entire path, including the reliable wired links.

One must tailor the existing TCP protocol for mobile environment without violating the end-to-end TCP semantics or the layered protocol stack approach. There are several proposals that specify TCP semantics in mobile networks. The Snoop protocol [Balakrishnan+ 97] relies on an agent at a base station to monitor the reliable delivery of link layer frames carrying TCP traffic. Does this solution stay true to the layered approach of solving networking problems? Should transport layer information be available to the data link layer? Indirect TCP (I-TCP) [Bakre+ 95] relies on establishing two connections for a TCP connection. One connection spans the relatively reliable and bandwidth-rich wired part; the other spans only the wireless link and is optimised for this link. In some ways, I-TCP does violate the end-to-end semantics of TCP connections.

Ideally, intermediate nodes should treat all the packets the same, whether they carry TCP or UDP (User Datagram Protocol) traffic. So, how widely deployable is a solution that expects intermediate nodes to do special processing for TCP traffic? Will such a solution work for TCP connections that span several diverse networks, all of which might not be under one administrative control?

For MANETs, the network's dynamism and the resultant changes in path length and propagation time can impact on TCP's mechanisms for congestion control in ways that are difficult to predict. So, are TCP's simple time-out and self-clocking mechanisms good enough to handle such dynamism? Or do we need new solutions for end-to-end traffic management in a MANET?

5. Addressing Issues: Different types of mobile devices exist such as laptop, palmtop, cellular telephones etc. These entities have different properties and necessitate dynamic bindings between their addresses and names. These bindings also change frequently because of global roaming properties of the mobile devices. For instance, within a network, a mobile terminal may transition from the
PSTN to the Internet, from the Internet to Asynchronous Transfer Mode (ATM) networks, from ATM to satellite, or from/to any other combination thereof [Akyildiz+ 99]. The critical issue then is to develop a transport method that makes these diverse technologies transparent to the application. The transport method will be required to track crucial mobile characteristics, such as addresses and identification. The use of well defined and standardised user/terminal characteristics are needed to manage the following operations [Pandya+ 97]:

- Determination of the home network or database of a roaming terminal.
- Identification of a mobile terminal on the radio control path for update and registration.
- Identification of the mobile terminal for terminal paging and location advertisement.
- Identification of the mobile terminal when exchanging location or routing information between different network types.
- Identification of the mobile terminal for retrieving the user profile.

6. **Mobility Support for Applications**: The challenges faced here are not only to design new applications but also to port existing applications to the mobile environment. The key issue that needs to be considered here is disconnected operation mode of mobile nodes and fluctuating bit rate. This may involve a partial disconnection, when the unit is in the doze mode or a total disconnection when the network connection is shut down by switching off the mobile device. A mobile node saves energy by avoiding transmissions and waking up from the doze mode only when absolutely necessary. The node is lazy in that it transmits only when it has to and it dozes off as often as it can. Voluntary disconnections can be treated as planned failures, which can be anticipated and prepared. Also there may be various degrees of disconnection ranging from total disconnection to weak disconnection. Weak disconnection or narrow connection occurs when a terminal is connected to the rest of the network via low bandwidth wireless channel. Even when a mobile node is connected, the data rates will vary considerably and will suffer from high error rates. An application developer must take into considerations these in his application definition. These issues make a larger impact on areas such as distributed data management, query and transaction processing.

Solutions based on existing cache-coherence and replica-consistency solutions in the context of mobile applications have been proposed. The issue here is as follows. Reads performed on the local copy are not guaranteed to return the result of the latest write. Because writes might be performed concurrently on mutually unreachable replicas, the replicas' state might diverge. So, when two previously unreachable nodes with mutually divergent replicas become reachable, how do you synchronize the replicas? On the other hand, it may be argued that insistence on strict replica consistency or cache coherence might be bad for the disconnected mode, as data availability will be adversely affected. So, a weaker notion of replica consistency is required. However, the meaning of ‘weak data consistency’ must be first defined.

A related issue is network-aware computing. As the mobile device moves between heterogeneous networks (from connection to a wired access point, to a wireless
connection through a wireless LAN, to a MANET) and as the speed of motion changes, the quality of the network link and of other available resources might change significantly. So, the application servers and clients should be able to adjust accordingly. For example, a user walks along a street carrying a mobile node with a high-bandwidth wireless link but with a limited energy supply and a very basic I/O interface. Later, the user enters a car, slips the mobile node into its docking port, and starts driving along a highway. Now, the node has a relatively abundant energy supply from the car's battery and a much better I/O interface, but a much lower wireless bandwidth owing to the car's high speed. What do the server and client need to do to hide such environmental fluctuations from the user? Should the applications be designed to adapt to changing conditions? Alternatively, should the data be stored in a variety of forms, each suited for a specific condition? These are some issues that need further examination.

7. **Database Issues:** Database storage and retrieval present several challenges in a mobile environment. For example, consider this scenario described in [Akyildiz+ 99]. A mobile terminal whose home network is a traditional PSTN is currently visiting a wireless ATM based (WATM) network. In order to register its current location with its home network, the terminal must be registered as a visitor on the network level with the ATM network, possibly requiring database update/query, and then the ATM network must send this updated profile to the mobile terminal’s home network for further database update. Call delivery from the PSTN to WATM will result in an increase in signalling overheads if the WATM network employs location advertisement and paging. This in turn will increase the number of database transactions, paging operations and broadcasts.

8. **Security Issues:** Security is a major issue in the mobile computing environment. The regular security threats that are prevalent in a traditional wired network also hold true in mobile computing environment. However, these threats are further aggravated in such networks. For instance, being accessible at any location and at any time creates concern about privacy issues for potential users [Imielinsky+ 94]. Appropriate policy and mechanisms must be in place that enforce rules pertaining to who is authorised to access a particular user and when and where do these authorizations take place. With mobile users, the problem of authentication will be at a global scale and distributed services and protocols to support authentication across administrative domains will be necessary. An orthogonal issue relates to anonymity. A user may like to have seamless connectivity while on the move but at the same time wish that its identity be concealed from visited networks. Security issues were largely ignored in the design of mobile networks. Of late, several proposals have come into existence but this area is still in its infancy.

9. **Issues relating to Standardization:** Standardization seems to be the slowest area of progress [Akyildiz+ 99]. While there is a general agreement among Japan, Europe, the US, and other countries that global standards are in everyone’s best interests, some difficulties still exist in achieving the necessary cooperation between regional and international bodies. In order to obtain the projected level of tetherless, global communication, each of these entities must be able to co-exist and cooperate within one infrastructure [Buchanan+ 97].
2.5 Summary

Mobile Computing and related networking issues have received considerable attention in the academic and research community. In this chapter we identified some fundamental concepts pertaining to mobile computing and identified its uses. Mobile computing offers new challenges that make an impact all levels of the communication protocol stack. In this chapter we identified these challenges that can be formulated into a number of open problems. Although mobile computing gives rise to several technical challenges, it also brings with it a promise for a great enhancement of human productivity.
Part B
Location Management and Security in Mobile IP
Chapter 3

Introduction to Mobile IP

3.1 Introduction

Regardless of the type of mobile network, one of the most important and challenging problems for seamless and tetherless communication is location management. In location management, an attempt is made to keep track of the mobile hosts in real time, as the hosts move through the network. The location records have to be updated frequently so as to keep the location management in real time. An efficient location management technique locates a user rapidly in a mobile network improving performance of the communication network. Such a technique, in general enhances the overall communication services in a mobile environment.

Location management is a two-stage process that allows the network to discover the current attachment point of a mobile host for message delivery [Akyildiz+ 99]. The two stages are location registration and message delivery. In the location registration stage, the mobile host must periodically inform some administrative entity in the network of its current point of attachment to the network. This may involve the administrative entity authenticating the user on behalf of the host and revising the host’s location profile. In the message delivery stage some correspondent entity queries the administrative entity for the host’s location and as a response to this the current location of the host is returned. An inherent feature of location management is handoff management. The purpose of handoff management is to enable the network to maintain a user’s connection as the mobile host continues to move and change its access point in the network. This usually involves three stages. Initially, the user, an administrative entity in the network or changing network conditions compels the participating entities to initiate the handoff process. In the second stage, the network must find new resources for the handoff connection and perform additional routing operations. Finally, the delivery of data from the old connection path to the new connection path is maintained according to agreed upon service guarantees.

The key questions that need to be addressed are:

- When should the mobile host inform the network of a change in its location?
- When a message arrives at a network administrative entity, how should this entity determine the exact location of the receiving mobile host within a specific time constraint?
- Who in the network must take the responsibility of managing the location information of a mobile host?
- Should this management framework be centralised or distributed? In other words, how should the user location information be stored and disseminated throughout the network?

While the first three questions pertain to location management algorithms, the last one deals with location information storage and database. Hence a technique for location management would involve the design of control messages between various
components and entities of the network along with a database architecture design for storage of location information.

Mobile networked computing is also raising important information security and privacy issues. The threats that exist in a traditional wired networked environment not only hold true for mobile hosts but these threats are further aggravated here. Also mobility introduces some new types of threats that are previously unheard of in static wired environment. The ultimate goal of a mobile network is to fulfil two general expectations:

1. To allow a mobile node to seamlessly access the network irrespective of its location in a secure way.
2. To protect the network from various attacks that can occur from malicious roaming hosts.

The purpose of this chapter is to provide an overview of the issues pertaining to location management and security in mobile networks. The focus is exclusively on mobility in the Internet. Since the Internet Protocol (IP) drives the Internet, an approach to facilitate mobility in the Internet must take IP into consideration. In general, this chapter is organised in the following way. Several schemes are in place that carry out location management on the basis of parameters such as time, distance and movement. Four major proposals addressing this issue have come from Internet Engineering Task Force (IETF) Mobile IP working Group [RFC2002], Columbia [Ioannidis+ 91], Sony VIP model [Teroaka+ 92], and IBM [Carlberg+ 91]. Out of these proposals the one from [RFC 2002] known as the IETF Mobile IP scheme has gained acceptance in the commercial environment. In this chapter, we first provide a brief overview of IP version 6 in section 3.2. In section 3.3, we describe the major Mobile IP proposals with an emphasis on the IETF mobile IP scheme. In section 3.4, we provide a brief survey of some other recent proposals in the area of location management. In section 3.5, we do a survey of schemes proposed so far to secure mobile IP. Section 3.7 we provide concluding remarks.

3.2 Next Generation: Internet Protocol Version 6 (IPv6)

For Basic information on TCP/IP and IPv4 routing, see Appendix A. IP version 4 uses class based two level address structure that leads to inefficient use of address space. As an example, if an organisation is assigned a class A address, 16 million host addresses from this address space are assigned for the organisation’s exclusive use. In this scheme of things millions of addresses are wasted. Apart from this, real time audio and video applications have stringent requirements in terms of delay strategies and reservation of resources that is not accommodated in IPv4 design. Also security services such as encryption and authentication are not catered to in the basic IPv4 design. To overcome these deficiencies IPv6 (Internet protocol version 6) [RFC 1365] also known as IPng (Internetworking Protocol, next generation) was proposed and is now a standard.

Mobile computing can also be supported on IPv6. Additional options such as exchange of care of address in a secure fashion and a new binding update message to notify the correspondent nodes about the mobile host’s care of address improves performance. However in this thesis, we focus on IPv4 as the underlying transport
mechanism to support mobile IP. This is due to the fact that IPv4 is still very widely deployed and is the only networking protocol that underpins the entire Internet.

3.3 Routing in the Mobile Environment

IP addresses are used by the applications to identify routes by which packets may be exchanged between two network nodes, namely the nodes performing the actions needed for the application [Perkins 98]. On the other hand, an IP address by which packets may be exchanged between two network nodes is also used to identify the endpoints themselves. This dual use of the IP address by the application endpoints causes problems that are encountered when trying to use the application while changing one’s point of attachment to the Internet. Clearly, applications need a fixed way to identify the communication endpoints, but just as clearly the routes between the endpoints must change as they move from place to place within the Internet.

3.3.1. IETF Mobile IP Architecture

Mobile IP is an open standard, defined by the Internet Engineering Task Force (IETF) RFC 2002, that allows users to keep the same IP address, stay connected, and maintain ongoing applications while roaming between IP networks. Mobile IP is scalable for the Internet because it is based on IP — any media that can support IP can support Mobile IP.
The number of wireless devices for voice or data is projected to surpass the number of fixed devices [Cisco 2002]. Mobile data communication will likely emerge as the technology supporting most communication including voice and video. Mobile data communication will be pervasive in cellular systems such as 3G and in wireless LAN such as 802.11, and will extend into satellite communication. Though mobility may be enabled by link-layer technologies, data crossing networks or different link layers is still a problem. The solution to this problem is a standards-based protocol, Mobile IP.

Mobile IP was created to enable users to keep the same IP address while travelling to a different network (which may even be on a different wireless operator), thus ensuring that a roaming individual could continue communication without sessions or connections being dropped. Because the mobility functions of Mobile IP are performed at the network layer rather than the physical layer, the mobile device can span different types of wireless and wire-line networks while maintaining connections and ongoing applications. Remote login, remote printing, and file transfers are some examples of applications where it is undesirable to interrupt communications while an individual roams across network boundaries. Also, certain network services, such as software licenses and access privileges, are based on IP addresses. Changing these IP addresses could compromise the network services.

Mobile IP has the following three components, as shown in Figure 2:

- Mobile Node
- Home Agent
- Foreign Agent

![Figure 2: IETF Mobile IP Architecture](image)

The Mobile Node is a device such as a cell phone, personal digital assistant, or laptop whose software enables network-roaming capabilities. The Home Agent is a router on the home network serving as the anchor point for communication with the Mobile Node; it tunnels packets from a device on the Internet, called a Correspondent Node, to the roaming Mobile Node. (A tunnel is established between the Home Agent and a reachable point for the Mobile Node in the foreign network.) The Foreign Agent is a router that may function as the point of attachment for the Mobile Node when it
roams to a foreign network, delivering packets from the Home Agent to the Mobile Node.

The care-of address is the termination point of the tunnel toward the Mobile Node when it is on a foreign network. The Home Agent maintains an association between the home IP address of the Mobile Node and its care-of address, which is the current location of the Mobile Node on the foreign or visited network.

Mobile IP process has three main phases, which are discussed in the following sections.

- Agent Discovery:
  A Mobile Node discovers its Foreign and Home Agents during agent discovery.

- Registration
  The Mobile Node registers its current location with the Foreign Agent and Home Agent during registration.

- Tunnelling
  A reciprocal tunnel is set up by the Home Agent to the care-of address (current location of the Mobile Node on the foreign network) to route packets to the Mobile Node as it roams.

**Agent Discovery**

During the agent discovery phase, the Home Agent and Foreign Agent advertise their services on the network by using the ICMP Router Discovery Protocol (IRDP). The Mobile Node listens to these advertisements to determine if it is connected to its home network or foreign network.

The IRDP advertisements carry Mobile IP extensions that specify whether an agent is a Home Agent, Foreign Agent, or both; its care-of address; the types of services it will provide such as reverse tunnelling and generic routing encapsulation (GRE); and the allowed registration lifetime or roaming period for visiting Mobile Nodes. Rather than waiting for agent advertisements, a Mobile Node can send out an agent solicitation. This solicitation forces any agents on the link to immediately send an agent advertisement.

If a Mobile Node determines that it is connected to a foreign network, it acquires a care-of address. Two types of care-of addresses exist:

- Care-of address acquired from a Foreign Agent
- Co-located care-of address

A Foreign Agent care-of address is an IP address of a Foreign Agent that has an interface on the foreign network being visited by a Mobile Node. A Mobile Node that acquires this type of care-of address can share the address with other Mobile Nodes. A co-located care-of address is an IP address temporarily assigned to the interface of
the Mobile Node itself. A co-located care-of address represents the current position of the Mobile Node on the foreign network and can be used by only one Mobile Node at a time. When the Mobile Node hears a Foreign Agent advertisement and detects that it has moved outside of its home network, it begins registration.

Registration

The Mobile Node is configured with the IP address and mobility security association (which includes the shared key) of its Home Agent. In addition, the Mobile Node is configured with either its home IP address, or another user identifier, such as a Network Access Identifier.

The Mobile Node uses this information along with the information that it learns from the Foreign Agent advertisements to form a Mobile IP registration request. It adds the registration request to its pending list and sends the registration request to its Home Agent either through the Foreign Agent or directly if it is using a co-located care-of address and is not required to register through the Foreign Agent. If the registration request is sent through the Foreign Agent, the Foreign Agent checks the validity of the registration request, which includes checking that the requested lifetime does not exceed its limitations, the requested tunnel encapsulation is available, and that reverse tunnelling is supported. If the registration request is valid, the Foreign Agent adds the visiting Mobile Node to its pending list before relaying the request to the Home Agent. If the registration request is not valid, the Foreign Agent sends a registration reply with appropriate error code to the Mobile Node.

The Home Agent checks the validity of the registration request, which includes authentication of the Mobile Node. If the registration request is valid, the Home Agent creates a mobility binding (an association of the Mobile Node with its care-of address), a tunnel to the care-of address, and a routing entry for forwarding packets to the home address through the tunnel.

The Home Agent then sends a registration reply to the Mobile Node through the Foreign Agent (if the registration request was received via the Foreign Agent) or directly to the Mobile Node. If the registration request is not valid, the Home Agent rejects the request by sending a registration reply with an appropriate error code. The Foreign Agent checks the validity of the registration reply, including ensuring that an associated registration request exists in its pending list. If the registration reply is valid, the Foreign Agent adds the Mobile Node to its visitor list, establishes a tunnel to the Home Agent, and creates a routing entry for forwarding packets to the home address. It then relays the registration reply to the Mobile Node.

Finally, the Mobile Node checks the validity of the registration reply, which includes ensuring an associated request is in its pending list as well as proper authentication of the Home Agent. If the registration reply is not valid, the Mobile Node discards the reply. If a valid registration reply specifies that the registration be accepted, the Mobile Node is assured that the mobility agents are aware of its roaming. In the co-located care-of address case, it adds a tunnel to the Home Agent. Subsequently, it sends all packets to the Foreign Agent.
The Mobile Node should register before its registration lifetime expires. The Home Agent and Foreign Agent update their mobility binding and visitor entry, respectively, during re-registration. In the case where registration is denied, the Mobile Node makes the necessary adjustments and attempts to register again. For example, if the registration is denied because of time mismatch and the Home Agent sends back its time stamp for synchronization, the Mobile Node adjusts the time stamp in future registration requests. Thus, a successful Mobile IP registration sets up the routing mechanism for transporting packets to and from the Mobile Node as it roams.

**Tunnelling**

The Mobile Node sends packets using its home IP address, effectively maintaining the appearance that it is always on its home network. Even while the Mobile Node is roaming on foreign networks, its movements are transparent to correspondent nodes. Data packets addressed to the Mobile Node are routed to its home network, where the Home Agent now intercepts and tunnels them to the care-of address for the Mobile Node. Tunnelling has two primary functions: encapsulation of the data packet to reach the tunnel endpoint, and decapsulation when the packet is delivered at that endpoint. The default tunnel mode is IP Encapsulation within IP Encapsulation. Optionally, Generic Record Encapsulation (GRE) and minimal encapsulation within IP may be used.

Typically, the Mobile Node sends packets to the Foreign Agent, which routes them to their final destination, the Correspondent Node, as shown in Figure 3.

![Figure 3: Tunnelling Procedure in IETF Mobile IP](image)

However, this data path is topologically incorrect because it does not reflect the true IP network source for the data, rather, it reflects the home network of the Mobile Node. Because the packets show the home network as their source inside a foreign network, an access control list on routers in the network called ingress filtering drops the packets instead of forwarding them. A feature called reverse tunnelling solves this problem by having the Foreign Agent tunnel packets back to the Home Agent when it receives them from the Mobile Node. (See figure 4).
Figure 4: Reverse Tunnelling in IETF Mobile IP

Security

Mobile IP uses a strong authentication scheme for security purposes. All registration messages between a Mobile Node and Home Agent are required to contain the Mobile-Home Authentication Extension (MHAE).

A pre-shared 128-bit key between a Mobile Node and Home Agent protects the integrity of the registration messages. The keyed message digest algorithm 5 (MD5) in "prefix-suffix" mode is used to compute the authenticator value in the appended MHAE, which is mandatory. Mobile IP also supports the hash-based message authentication code (HMAC-MD5). The receiver compares the authenticator value it computes over the message with the value in the extension to verify the authenticity.

Optionally, the Mobile-Foreign Authentication Extension and Foreign-Home Authentication Extension are appended to protect message exchanges between a Mobile Node and Foreign Agent and between a Foreign Agent and Home Agent, respectively.

Replay protection uses the identification field in the registration messages as a timestamp and sequence number. The Home Agent returns its time stamp to synchronize the Mobile Node for registration.

Some Issues

There are some issues concerning binding in mobile IP. For instance, a mobile host can maintain several care of Addresses at one time. This implies that a Home Agent must tunnel packets to all these care of addresses resulting in duplicate datagrams arriving at the target mobile host. A mobile host must have mechanisms to filter out these duplicate packets. Due to slow incorporation of wireless LANs, simultaneous binding has not yet been made available [Perkins 97].

Currently, three major concepts have been identified as potential methods for limiting location update and registration cost in Mobile IP [Akyildiz+ 99]. Firstly there should be mechanisms available that can facilitate local connectivity for mobile hosts and manage the buffering of datagrams to be delivered as this may result in smooth handoffs. Secondly, there is a need to integrate a multicast facility into this
architecture that will allow the mobile host to use the services of IP multicast especially when it is residing in a foreign network. Finally, a hierarchy of foreign agents can be deployed in agent advertisement messages in order to localise the registration to the lowest common FA of the CoA at the two points of attachments. These are some of the issues that need more investigation.

### 3.3.2 Other Prominent Mobile IP Schemes

The three schemes mentioned above are described here.

#### 3.3.2.1 Columbia Mobile IP

Researchers at Columbia University were among the first to begin experiments in mobile networking. They aimed to provide campus mobility for mobile nodes, partially as an outgrowth of the Student Electronic Notebook (SEN) Project. The basis of Columbia mobile IP [Ioannides+ 91, Ioannidis+ 93b] is the definition of a virtual mobile subnet that is created by placing a small number of cooperating mobile subnet routers (MSRs) wherever a MH may be connected to the network. The collection of mobile support routers (MSRs) collaborate to create a mobile subnet that appears to the rest of the network as a real subnet connecting existing infrastructure by MSRs acting as routers. As mobile nodes move, they detect beacons emitted by the MSRs according to the Mobile Internet Control Protocol [MICP], comparable to ICMP. As mobile nodes move from place to place, they inform their current MSR about their needs and requested the current MSR inform their previous MSR of their movement. The previous MSR caches the mobile host’s current location. In this way, all MSRs can remain up to date regarding the movement of the mobile node.

A mobile host is assigned a constant address on the mobile subnet. This means higher layer protocols have an unchanging view of the mobile host’s identity. The fact that the mobile subnet is spread across a number of real subnets gives the MH its mobility. The MSRs communicated by way of a new multicast address, which they had to join. Figure 5 shows an illustration of the Columbia protocol.

Since the mobile subnet was effectively a virtual subnet, no support for existing hosts on the mobile subnet was necessary. The Columbia protocol also served as the basis for numerous other research efforts into mobile networking at Rutgers and Brown universities, as well as further efforts at Columbia during the early 1990s, especially important first attempts to provide network-layer security [Ioannides+ 93b]. Concerns about scalability, however drove attempts to avoid distributing functions to a variable population of symmetric agents maintaining location information for the mobile nodes. With centralized location information, the use of multicast was no longer warranted, gaining possible further improvements in scalability.
3.3.2.2 Sony: Virtual Internet Protocol

This proposal aims to provide host migration transparency to the transport layer. To accomplish this, it introduces two network layer identifiers to the host: one is migration independent (VN or VIP address) and the other is migration dependent (PN or Temporary IP [TIP] address.). The transport layer can specify the target host by the PN address and therefore is not aware of host migration. An IP address can be considered a PN address since it specifies the location of a host in the Internet.

When a mobile host connects to a new location it must first acquire a temporary address from a local address server. This address represents the current network location of the mobile host. If the mobile host is connected to its home network then the VIP and TIP are equal. A MH always informs a gateway attached to its home network of its current TIP.

The division of the location and identification information into two addresses requires extra space within the packets. The Sony scheme defines a new IP option to carry the identification information (source and destination VIPs) while the address fields in the packet header carry the location information (source and destination TIPs). The VIP option shown in the figure also carries packet type; hold time, and time stamp information.

The Sony scheme defines an efficient method of distributing the VIP to TIP mapping called the propagating cache method. In the propagating cache method, every host and gateway has a cache for address conversion, called an AMT (Address Mapping Table).
If the source host has an AMT entry for the destination host, the source host executes address conversion before sending the packet; the TIP sublayer can then correctly deliver this packet to the destination host. On the other hand, if the source host has no AMT entry for the destination host, the source host assumes that the destination host is connected to its home network and sends the packet accordingly.

As the packet traverses the network in transit to the home network of the destination host, if an intermediate gateway has the AMT entry for the destination host, the gateway executes address conversion and forwards the packet to the current physical location of the destination host. When a host or gateway receives a packet, the VIP sub-layer creates or updates the AMT entry for the source host of the received packet. Thus, the AMT information propagates across the interconnected networks as communication progresses.

In the case when there are no intermediate gateways that know the current mapping of the mobile host, the packet will eventually arrive at the mobile host’s home network. If the target mobile host is connected to the home network, the packet will be delivered or else, the home gateway will modify the destination IP address field to reflect the current location mobile host’s current location and retransmit the packet. Subsequently, when the sending node receives a packet back from the target mobile host, it will store the mapping contained within the packet in its own AMT. All the future packets to this sender to the target mobile host will then be sent directly to the current location of target mobile host through the most efficient route. Intermediate gateways will also cache these mappings, which will assist in the efficient delivery of packets from other hosts to this mobile host.

The AMT entry is held until it becomes obsolete, invalid, timed-out or updated. The home gateway sends a special management packet called VipDelAmt to invalidate a packet when a mapping changes. The invalidation process is not guaranteed and when invalid entries are detected a ViperrObs packet is issued to obsolete the erroneous cache entries. Cache entries are either timed out using a timer VIPhold or updated continuously by passing packets.
This model is scalable in the sense that the number of entries, which a host or a gateway must have, is independent of both the scale of the interconnected networks and to the total number of migrating hosts. It is sufficient for a host to have AMT entries for other hosts with which the host communicates simultaneously. A host never creates an AMT entry for unrelated hosts. A further feature of the Sony model is its definition of compatibility mode to ensure that mobile hosts can communicate with regular hosts (using existing protocols). The compatibility mode swaps the source IP and VIP in the packet so that a regular host always receives the VIP as the source address.

3.3.2.3 IBM Mobile IP scheme

The IBM Mobile IP scheme uses the services of the transport layers TCP and UDP to provide mobility related services. In order to accomplish this task, they employ the loose source routing (LSSR) option specified in the IP header. In [Braden 89] it is stated that if TCP receives a packet in LSSR option, then any reply sent must take the same LSSR in reverse. In the case of UDP, the application residing on top of UDP uses the reverse LSSR path. These features are used to enable existing hosts to participate in the routing to mobile hosts.

The IBM mobile IP scheme works as follows. When a mobile host (MH) visits a foreign subnet it registers itself with a local base station (BAS) representing the foreign subnet. It also informs its home station, called the Mobile Router (MR), of the specifics of its current location, including the address of BAS. If some correspondent host transmits a packet to this MH, the packet using the existing Internet routing infrastructure arrives at MR representing MH. MR refers to its database to identify the current location of MH and inserts a LSSR in the packet with MH’s current BAS as the first hop. The packet is forwarded using existing routing infrastructure facilities to the MH. MH specifies LSSR in its return traffic with BAS as its first hop. This way, the returned traffic forces the subsequent packets from correspondent host to be routed directly to the MH. This process is illustrated in the figure 7.

Upon migrating to a new subnet, MH must repeat the same process of registering with a local BAS and must also request this BAS to notify its previous BAS to delete its location entries. Subsequent packets from the correspondent node to the MH will be redirected by the previous BAS back to the MR for correct routing until a return packet from the MH to the correspondent host forces the correspondent host to route the packets correctly via the new BAS. Communication between two MHs is achieved using a similar process. The only difference is that the LSSRs will specify two hops, namely the BASs with which each MH is registered. The IBM proposal is not a novel mobile IP specification as such; rather, it is more of a general specification that encompasses ideas of other mobile IP proposals.
3.4 Other related Work in Location management

Besides the proposals described above, there are numerous other location management schemes for mobile IP proposed in recent years. An early work was done by [Fowler 86]. This proposal described a technique for efficiently forwarding pointers for finding decentralised objects. This idea was borrowed in a technique proposed by [Krishna+ 96].

For on-line tracking of mobile hosts a model is proposed by [Awerbuch+ 91]. The architecture consists of m-regional matching directories. Localised updates and searches are carried out using forwarding pointers and regional matching directories.

The proposal from [Spreitzer+ 93] describes a network architecture that consists of user agents and a location query service. Each user has a dedicated user agent that is responsible for forwarding any communication to or from the user. This scheme is suited for local networks constrained within a building’s premises. The major disadvantage with this scheme is that as the number of hosts in the network increases, it is not efficient to have a dedicated user agent per user.

The proposal to cache data at the Internet Access Point (IAP) comes from [Wu+ 93]. The idea is that in order to carry out optimum routing it is best to cache location data at IAP. If IAP does not have an entry for a host, the message is forwarded to the mobile router (MR), which maintains information for all hosts. This scheme is effective for local networks but as the network size increases, the MR becomes a bottleneck and perhaps a single point of failure.
A different approach to solving the location management problem comes from [Badrinath+ 92]. The strategy here is to use a technique to carry out location management such that the search costs are reduced and also the volume of location updates is controlled by employing user profiles. This model defines a hierarchy of location servers that are interconnected and are also connected to base stations (or mobile support stations) by a static wired network. Partitions are created by using user profiles. It is only when a user crosses a partition, that the location update takes place. The problem with this scheme is that profiles are not always available a priori. Other similar techniques such as user profile replication in [Shivakumar+ 95] and local anchoring [Ho+ 95] have also been proposed to reduce network load due to location management.

A modified tree structure for location management is proposed in [Dolev+ 95]. The root and some of the higher levels of the tree is replaced by a set-ary butterfly network. This helps in reducing search requests at the nodes. This proposal also suggest schemes that make the protocol self-stabilizing.

The idea of location management using home location servers has been proposed in [Jain+ 94] and [Jain+ 95]. These schemes independently propose caching [Jain+ 94] and forwarding [Jain+ 95] for Personal Communication services (PCS). A similar scheme is suggested by [Krishna+ 96]. This scheme proposes forwarding techniques that augment the IS-41 scheme [IS95] to provide efficient location management. The search cost is reduced using a technique called search-updates. The scheme suggested in [Krishna+ 96] combines forwarding and caching schemes to improve performance when compared with a scheme that uses only forwarding as in [Jain+ 96]. However, combining caching and forwarding brings about unique problems with forwarding pointer maintenance. To eliminate this problem [Krishna+ 96] have also proposed a method forwarding pointer maintenance and they also take into account the overhead maintenance and related costs. The maintenance and associated cost is not taken into account in [Jain+ 95].

3.5 Secure Location Management in Mobile Networks

Mobile nodes unlike their counterparts, the static nodes, do have some problems in getting access to certain services that they require. This is due to several factors highlighted above such as low power availability as the mobile nodes on the move use battery power, low computation resources due to their compact nature and low bandwidth for the wireless links. Apart from these limitations, which can be upgraded in future, they are faced with difficult problems of security as they move from one domain to another. At the same time, the administrative entities that manage the mobile networks are also vulnerable to hostile attacks from unknown and malicious mobile nodes. While security threats that occur in a traditional wired networks also hold true for mobile networks, they are further aggravated in this environment. Mobility also introduces new security vulnerabilities to the network.

In this section we will first identify the threats that can occur in networked systems with an emphasis on mobility and then identify the corresponding services that are required to counteract such threats. We will then examine some schemes that aim to provide security for mobile networks.
Security Threats and Services in Mobile IP networks

1. **Masquerading:** In this type of an attack, one attacker pretends to be another entity. As a consequence, the attacker can get hold of privileges, which it is not authorised to have in the first place. Over the network, for instance the attacker can pretend to be legitimate client to gain access to a server.

2. **Unauthorised use of resources:** In this type of attack, the attacker in an unauthorised way gains access to resources on the network. For instance, the threat may be in the form of accessing a simple resource as a printer or it can be more complex such as a database, or some applications within the database. Such an unauthorised access may lead to the destruction, modification, and disclosure of information critical to an individual or an organisation.

3. **Unauthorised disclosure and Flow of Information:** This threat involves unauthorised disclosure or illegal flow of information stored, processed, or transferred in a networked system, both internal and external to the user’s organisation. In a network, this attack normally manifests itself in passive wiretapping or traffic analysis.

4. **Unauthorised Alteration of Resources and Information:** This form of attack may occur over the network in the form of active wiretapping. In some cases, this attack may occur in combination with other attacks such as replay whereby a message or part of a message is repeated intentionally to produce an unauthorised effect. This threat may also involve unauthorised introduction and removal of resources into or from a networked system.

5. **Repudiation of Actions:** This is a threat against accountability. It occurs when a sender or receiver of a message denies having sent or received the information. For instance, a customer engages in a transaction with a bank to withdraw a certain amount from his account, but later denies having sent the message. A similar attack can occur at the receiving end. For instance, a firm denying the bid offer for the tender even though it actually did receive that offer.

6. **Unauthorised denial of Service:** Here, the attacker, acts to deny resources or services to entities, which are authorised to use them. In the networked system, often the attacker blocks the access to a network resource by continuous deletion or generation of messages so that the target resource is either depleted or saturated with meaningless messages.

Having identified all the threats, it is also essential to understand that for a particular environment, one needs to determine which threats are applicable. The overall set of security measures required to counteract the identified threats constitutes the security policy. To be effective, the security measures need to be coherent and complete including physical security (e.g. vaults and door locks), procedural security (e.g. selecting personnel with regard to trustworthiness and logical security. The security architecture described in this thesis only addresses the logical security.

Counteracting the relevant security threats involves provision and enforcement of appropriate security services and mechanisms. Any security architecture should be able to support a wide range of systems and applications, and consequently it is intended that it should support a wide range of security services that can be used and combined in different ways to meet different security policies. In particular, a distributed environment is likely to have multiple security policies and different
authorities responsible for various parts of the system. In this section we will describe in general terms the security services that are required to meet the above mentioned threats in a distributed environment. In developing a security service, we need to address at least the following questions:

- What are the security attributes/information used by the service?
- What mechanisms can be used to provide the service and what are associated rules of operation?
- What are the authorities that are involved in the management of the service and its associated mechanisms?

We will now treat fundamental types of security services and describe the mechanisms that can be used to provide them. In general, to counteract the denial of service attack additional mechanisms over and above security mechanisms are required. We will not be considering this service separately in this thesis, though we mention how the other security services and mechanisms can help to detect some such attacks thereby reducing the degree of severity of this threat.

**Data Confidentiality Service:** A Confidentiality service helps to counteract the unauthorised disclosure of information attack by transforming the information using a key dependent cryptographic function, which allows only the authorised users with the right key to be able to read the original plaintext information. The mechanisms that are commonly used to provide confidentiality are based on cryptographic techniques. In an operating system environment, it may be sufficient to protect the confidentiality of information just by using access control mechanisms. However in a network environment this will not suffice, as there is a flow of information in plaintext format between two remote entities. Encryption techniques are used to achieve confidentiality. There are two commonly used encryption schemes, namely, the symmetric scheme and the asymmetric scheme.

In a symmetric encryption system, both the sender and the receiver use the same key to perform encryption and decryption operations. Consequently, in such a system, it is necessary to transfer the key between the two ends. If this shared key between the sender and the receiver is protected, both the secrecy and the authenticity of information are achieved. The most well known symmetric algorithm is the data Encryption Standard. It is a complex non-linear block cipher algorithm which transforms 64 bit input plaintext blocks to produce 64 bit cipher text blocks using a 56-bit key.

In an asymmetric or public key system, the enciphering and deciphering keys (known as the public and the private keys respectively) differ in such a way that at least one key is computationally infeasible to determine from the other. Thus one of the keys can be revealed without endangering the other. Protecting the separate keys, namely the decrypting key for secrecy and encrypting key for authenticity provides secrecy and authenticity of information. Hence in this case, in addition to the encryption and decryption procedures, the encryption keys can be made public. The receiver only makes the decryption key secret. Therefore anyone wishing to communicate with the receiver can do so using the publicly available key and only the intended receiver would be able to understand the information. The use of such systems thus avoids the necessity to transmit the key used by the algorithm over a secure channel among the
communicators. Moreover, such systems can be used to transmit the secret key required for asymmetric systems. The most well known asymmetric cryptosystem is the Rivest-Shamir-Adleman system, commonly known as RSA [Rivest+ 78]. It is based on the difference in computational difficulty between finding large prime numbers and factoring large numbers. The decryption process requires the knowledge of the prime factors of a large number whereas the encryption process does not. If the composite number is large enough (say 250 digits long), then the difficulty one would experience in factorizing it to its prime constitutes the security of the system.

Data Integrity Service: An unauthorised modification attack may involve unauthorised insertion and deletion of information transferred over the network. This attack often occurs in conjunction with other attacks such as replay whereby a message or part of a message is repeated intentionally to produce an unauthorised effect. An integrity service provides for the protection of information from unauthorized modification, that is, alteration, insertion, or deletion of information. For instance, within a system, this could be alteration of the contents of file, whole in the case of a network; this may be modification of data transferred over communication lines. The provision if integrity service comprises
1. The generation of integrity checks (at the originating end),
2. The verification of integrity checks (at the receiving end).

Typically, integrity mechanisms employ cryptographic techniques to produce integrity checksums which can be used to determine whether there has been any insertion, deletion or reordering of information. This is done by using feedback-chaining techniques, which make an output dependent not only on the current input and key but also on the earlier input(s) and/or output(s). An important use of such chaining techniques is in the detection of any alteration to information. This is because successive pieces of the information will cause an error in the decryption process leading to detection. The cipher block chaining (CBC) technique is the most common one used to generate an integrity checksum, sometimes referred to as the message authentication code (MAC).

Authentication and Identification Service: In a masquerading attack, an entity pretends to be another and attempts to gain privileges and access to information and resources to which it is not authorised. This service is used to counteract such masquerade attacks. Identification associates an identifier with an entity whereas authentication establishes the validity of the claimed entity. Typically authentication methods rely on some combination of the following:

- Something known to an entity, such as secret password, key and/or
- Possession of something physical, such as a card containing several details and/or
- Some immutable characteristic of a user, such as retinal scan and/or
- Context, for example the address of the entity and/or
- Accepting that an identified trusted third party has already established authenticity.

Authentication can be one-way or mutual. One-way authentication provides one party with assurance of the other’s identity but not vice versa. An example of this is data
origin authentication, which is used to prove that the message originated from the claimed source. Mutual authentication or Peer-entity authentication provides both parties with assurance of each other’s identity. In a more general situation, when two parties wish to authenticate each other, they may need to involve one or more third parties. A simple model is when we have a single trusted third party. In this case, authentication can be achieved as follows: The trusted third party first authenticates the claimant (the sender) and then proves the authenticity of the claimant to the verifier. The nature of trust between each party and the third party is an important issue in determining the assurance of service. Examples of trusted third parties include authentication servers, key management servers and certification authorities. Authentication mechanisms include the use of public key and symmetric key based techniques and Challenge-Response mechanisms. Nonces and timestamps can be used in authentication protocols to provide timeliness and freshness.

**Access Control Service and Authorization:** Another common attack is unauthorised access to network resources and services. Having successfully masqueraded as another entity, an entity can gain access to resources, which are otherwise denied to it. This service provides the ability to limit and control access to host systems, applications and information, and to limit what entities might do with the information contained, e.g. in applications, files. In an access control scheme, certain entities (initiators) attempt to access other entities (targets). An access control policy is essentially a set of rules that define the conditions under which initiators may access targets. The decision is to grant or deny a particular request is determined using access control rules and the access control information associated with the

With the introduction of mobility, some of the above mentioned threats get further aggravated.

- Wireless links are more vulnerable to attacks when compared to the traditional wired medium. A number of examples of cellular snooping are reported in [Neumann+ 95] and the American Congress has ruled that there is no expectation of guaranteed privacy in cellular telephone networks, which broadcast their signals widely in an unencrypted form.
- A mobile node can move across several different domains where a domain can be defined as a region under a single administrative authority. For instance, in the mobile IP case, the node may first initiate its journey when under its home agent, and subsequently end up roaming across several domains each of which is governed by a different foreign agent. In such a situation, strong authentication procedures need to be deployed to detect and thwart potential attacks on both the administrative entities such as foreign agents as well as the mobile hosts.
- Being reachable at any location and at any time creates greater concern about privacy issues among the potential users [Varadharajan+ 95]. For instance there may be a need for developing profiles, which specify who, when and from where one is authorised to get a service. One way is to provide mechanisms that restrict the list of users who are allowed to use a mobile appliance to send a message. From the management point of view, we need to address where are these profiles stored and how are they to be managed.
- Due to the roaming tendency of a mobile host, it may end up using resources at various locations. Each of these locations, as mentioned above, might map on to different administrative domains. In some cases, several administrative domains
must collaborate to provide a particular service to a mobile host. For instance, consider a mobile user who is a member of a stock mailing list that provides updates on stock market situation. This is a typical example of a multicast service. To make the system scalable, often the home agent in the home network of the mobile host also becomes a member of this multicast group. But when the mobile host moves into a foreign network, it might request this service either from the foreign agent or it might request the foreign agent to provide the same in cooperation with the mobile host’s home agent. In the former case, the foreign agent becomes the member of the multicast group and receives packets on behalf of the mobile host. In the latter case, the packets are tunnelled from the mobile host’s home agent to the current foreign agent of the mobile host who in turn delivers these packets to the mobile host. In such cases, it is necessary to define a trust relationship across different administrative domains.

Key questions that arise are: should we have a uniform trust policy across the domains or should the trust policy define different guidelines or levels of trust based on the identity of mobile hosts or on the basis of services that they wish to avail.

- Mobile devices themselves are highly vulnerable to physical and logical attacks as the owner of this device is constantly on the move if by any chance the device is left unattended, the attacker it could lead to theft of the device or loss of information stored in the device. Therefore, integrity and confidentiality of information stored on the mobile device is an important cause for concern.

- User anonymity is a matter of growing importance in business and private environments. Consider for example a scenario presented in [Fasbender+ 96] where takeover negotiations are on between two companies. In such cases, an indication of management meetings to outsiders would uncover strategic alliances. One might also think of burglars being able to discover that someone has left the house domain for a sufficiently long time.

- Finally an issue that is tightly coupled with anonymity is the privacy with respect to location information. For example, obvious misuse of location information by third parties is networks for military and police usage, where the knowledge of the location of network users is nearly as useful contents of transmitted messages themselves. There are several security related issues associated with revealing location information or propagating false or unauthenticated location information.

False location information can corrupt directories and routing tables, leading to misrouting of confidential information, hijacking of connections, and denial of service (DoS) because honest nodes cannot communicate. Various security mechanisms must be in place to thwart such attacks. Unauthenticated location information makes it possible for an attacker to misinform correspondents about the mobile host’s location and, thus, to redirect packets intended for the mobile host to a wrong destination. This can lead to the compromise of secrecy and integrity as well as denial-of-service because the target nodes are unable to communicate.
When sending false location binding messages, the attacker pretends to be a mobile host by masquerading itself as the host and using its IP address in its communication. It can initiate such an attack by either highjacking existing connections or by opening new ones. The attacker can also cause another type of denial of service attack by redirecting the packets to a random or non-existent care-of address in order to disrupt communication with the mobile. This implies that the attacker can redirect these attacks are alarming because the attacker can be anywhere on the network and all Internet nodes are potential targets.

IP addresses of mobile nodes are indistinguishable from those of stationary nodes. The attacker can make any other node believe (this includes even stationary hosts) that a particular host is on the move by supplying a false care of address. In order to accomplish this, the attacker just needs to know the addresses of the two communicating hosts. Hosts that have well-known and permanent addresses are highly vulnerable to such attacks.

In principle, the nature of these above mentioned attacks differ. For instance, in order to sabotage an existing communication session, the attacker must wait till the host first attempts to establish such a connection and then try to sabotage using redirection techniques. Whereas on the other hand, attacker can initiate a false location-binding message whenever it wants to do so.

Security services like authentication, integrity and non-repudiation can only be accurately provided by using some form of key management infrastructure that facilitates the distribution/exchange of keying information amongst messages senders and receivers. One simple way of providing this facility is to manually load keying information into each node. This is acceptable for a small number of nodes but this scheme does not scale to a large network. There are some other basic IETF security protocols and IPSec based schemes that can be deployed to provide key management services.

In order to counteract these attacks, a strong authentication scheme coupled with end-to-end encryption of messages and integrity of payload is required. In recent years several proposals have surfaced, that aim to address these issues. In this section we shall examine some prominent proposals that aim to provide security services for mobile IP. These proposals are as follows:

- Standard Mobile IP Authentication [Perkins 98].
- Mobile IP Authentication with AAA infrastructure [Schafer+ 2001].
- IP Security ((IPSec) [Kent 98].
- A Public Key Based Secure Mobile IP [Zao+ 99].
- Secure Mobile IP using Firewall Support [Gupta+ 98].
- Use of IPSec in Mobile IP [Zao+ 97].
- The Mix Method [Chaum 81].
- The Non Disclosure Method [Fasbender+ 96].

Several schemes use the services of IP security. So we first provide an overview of the IP security scheme. We then describe the standard mobile IP authentication
scheme followed by authentication of mobile IP using AAA. We thereafter look into other proposals as identified above.

3.5.1 IP Security

The proposal from [Kent 98] describes security architecture for the Internet protocol known as IPSec. IPSec provides security services by using two security protocols, namely the Authentication Header (AH) protocol and the Encapsulating Security Payload (ESP) protocol. The services provided are Access Control, Connectionless Integrity, Data Origin Authentication, Rejection of replayed packets, Confidentiality and Limited Traffic Flow confidentiality. A key concept that appears in both the authentication and confidentiality mechanisms is the Security Association (SA). An association is a one-way relationship between sender and receiver that provides security services to the traffic being conducted on that channel. However, if peer-to-peer relation is needed, then two security associations are required. A security association is uniquely defined by the following three parameters

1. **Security Parameter Index (SPI):** A sender assigns a bit string to the Security Association (SA), which bears only local significance. The receiving system will understand what algorithms are to be used to process the received packet by looking at this bit string.

2. **IP Destination Address:** This is the address of the destination end-point of the security association. Security Protocol Identifier: This indicates whether the association is an AH or ESP security association.

Both AH and ESP support two modes of use: Transport and Tunnel mode. Transport mode provides protection primarily for upper layer protocols. The payload in this mode is the data that normally follows the IP header. The drawback of this approach is that it is possible to do traffic analysis. Tunnel mode provides protection to the entire IP packet. To achieve this, the entire packet plus the security fields that are in the AH or ESP headers are treated as the payload of the new outer IP packet. As this outer packet may have a totally different source and destination addresses it appears to travel through a tunnel. With Tunnel mode, a number of hosts on networks behind firewalls can engage in secure communication without implementing IPSec.

The authentication header is used in transport mode when both the communicating ends share a protected secret and the authentication process is secure. However, if the communicating parties are behind the firewalls, or when the communicating parties rely on some other third party at both the ends to provide security owing to the lack of support for authentication feature, they choose the tunnel mode. In tunnel mode, the middle parties authenticate each other and provide reliable data to the communicating parties behind them. In the case of transport mode AH using IPv4, the AH is inserted after the original IP header and before the IP payload. However, in the case of tunnel mode AH for IPv4 the entire original IP packet is authenticated and the AH is inserted between the original IP header and a new outer IP header. The inner IP header carries the ultimate source and destination address; the outer IP header may contain different IP addresses.

Transport mode ESP is used to encrypt and optionally authenticate the data carried by IP. For this mode using IPv4, the ESP header is inserted into the IP packet
immediately prior to the transport layer header and an ESP trailer is placed after the IP packet. If authentication is selected, the ESP authentication data field is added after the ESP trailer. The entire transport level segment plus the ESP trailer are encrypted. Authentication covers all the cipher text and the ESP header. Transport mode operation can be summarized as follows:

1. At the source, the block of data consisting of the ESP trailer plus the entire transport layer segment is encrypted and the plain text is replaced by its corresponding cipher text to form the IP packet for transmission. Authentication is added if this option is selected.
2. The packet is then routed to the destination. Each intermediate router needs to examine and process the IP header plus any plain text IP extension headers but does not need to examine the cipher text.
3. The destination node examines and processes the IP header plus any plain text IP extension headers. Then on the basis of the SPI in the ESP header, the destination node decrypts the message.

Transport mode operation provides confidentiality for any application that uses it and is also reasonably efficient adding little to the total length of the IP packet. However, the drawback of this mode is that it is possible to do traffic analysis.

Tunnel mode ESP is used to encrypt an entire IP packet. For this mode, the ESP header is prefixed to the packet and then the packet plus the ESP trailer is encrypted. This method can be used to counter traffic analysis. A new IP header is then appended to the entire packet carrying different addresses. Tunnel mode operation can be explained as follows:

1. The source prepares an inner IP packet with a destination address of the target host. This packet is then prefixed by an ESP header and post-fixed with an ESP trailer. Then the packet along with the ESP trailer is encrypted and authentication data may be added. The resulting block is encapsulated with a new IP header.
2. Each intermediate router needs to examine and process the outer IP header plus any IP extension headers.
3. The destination then examines and processes the outer IP header plus any outer IP extensions. Then on the basis of the SPI in the ESP header, the destination node decrypts the message. It then transmits the packet in the internal network.

IPSec is concerned with using established security associations (SAs) to exchange secure traffic through tunnels. IPSec assumes the SAs have been established before IPSec packets are exchanged. This implies that there must be some means of setting up SAs between the communicating parties called nodes or peers. This latter option is the responsibility of Internet Security Association Key Management Protocol (ISAKMP) and Internet Key Exchange (IKE) Protocol. ISAKMP defines procedures to establish, negotiate, modify, and delete security associations. It does not define the procedures for how the key exchange itself occurs. This latter option is defined in IKE to securely generate and distributed the key for authentication and encryption.
3.5.2 Standard Mobile IP Authentication

In [Perkins 98], an authentication process is described that involves a cryptographic hash value to be appended to the registration messages in the form of an extension field. The extension includes type field, which indicates what are the two nodes involved in the authentication process. The extension also includes a length field, which specifies the payload length of the extension. A security parameter index (SPI) field is included in the extension, which specifies the security context like the authentication algorithm to be used, its mode and the key used. The last part of the extension is an authenticator field, which is computed over the entire Mobile IP registration message.

A mobile node that wants to register at a foreign network listens to the agent advertisements in that network. After having received the advertisement, the registration is carried out in the following steps:

1. The mobile node sends the registration message to the foreign agent which includes flags, describing the protocol to be used, the requested lifetime for the registration, home address of the mobile node, address of mobile nodes home agent, the care of address, an identifier of this request, mobile nodes network access identifier (NAI), an authentication extension to be checked by home agent and optionally an authentication extension checked by foreign agent.

2. Upon reception of this message, the foreign agent checks for mobile node/foreign agent authentication extension, if present, it computes a foreign agent/ home agent authentication extension and sends the message to the home agent. This message includes the same message sent by the mobile node with an addition of an authentication extension of the foreign agent to be checked by the home agent.

3. The home agent checks the authenticity of the received message by checking the authentication extensions in the message. The home agent then creates a registration reply message which includes the mobile IP specific result, lifetime of the registration, address of the mobile node, address of the home agent, an identifier of the reply, network access identifier of the mobile node, the home agent/mobile node authentication extension and optionally home agent/foreign agent authentication extension.

4. The foreign agent eventually computes the optional foreign agent/mobile node authentication extension and sends the message to the mobile node.

5. Upon reception of this message, the mobile node checks the authentication extensions included in the message and assumes that it has successfully registered.

This scheme recommends that if absolute protection from traffic analysis is required, the mobile host can establish a bi-directional tunnel to its HA. Then the address of the communicating hosts are shielded from examination at intermediate routers, as original source and destination addresses are invisible or even may be sealed when encrypted tunnels are used [Montenegro+ 96]. However, the traffic between the two corresponding networks that are tunnel endpoints can still be observed, which may be sufficient for an attacker even if he does not know which stations are communicating.
3.5.3 Mobile IP Authentication with AAA infrastructure

This architecture as described in [Schafer+ 2001], defines an administrative domain that has multiple foreign agents under the jurisdiction of one or more local AAA servers. In case, this administrative domain is the home domain, then it may consist of several home agents under one or more AAA servers. Refer to figure 8.

![Diagram of Mobile IP AAA Trust Model]

**Figure 8: Mobile IP AAA Trust Model**

In this architecture, a static trust relationship exists between the following entities:

1. Mobile nodes and their home AAA servers.
2. Foreign agents and their local AAA servers.
3. Home agents and their home AAA servers.
4. AAA servers and one or more AAA brokers.
5. Various AAA brokers.
6. AAAL and AAAH servers.

The following dynamic trust relationships are also established using the static trust relationships.

1. Mobile nodes and their home agents.
2. Mobile nodes and their foreign agents.
3. Foreign agents and home agents, which are currently involved.
The reasons for these dynamic relationships are self-explanatory except for case 1. The dynamic relationship between a mobile host and its home agent occurs for load balancing purposes. AAAH server must balance load equally among HAs of the network. The protocol involving AAA servers can be explained as follows:

1. Foreign agents periodically send out agent advertisement messages. These advertisement messages include the Network Access Identifier (NAI) extension of the foreign agent along with a random number to counter replay attacks.
2. Upon receiving such a message, a mobile host first stores the NAI of the foreign agent. It then generates a registration message that contains the random number of the foreign agent along with the mobile node’s NAI. A signature is computed over this message using the mobile node’s private key. The AAAH server of the mobile host must verify this signature.
3. Upon receiving this registration message, the foreign agent creates an AAA mobile registration request message, which contains the mobile node’s request message and sends it to the home AAA server.
4. The local AAA server either indirectly forwards the message by the use of AAA brokers or directly sends this message to the home AAA server of the mobile node by evaluating the contained NAI.
5. The home AAA server checks the signature of the mobile node in the mobile IP registration message. Upon successful validation of this signature, it creates a home agent registration message containing the mobile node’s original mobile IP registration message, a session key for use between mobile node and home agent, as well as a session key for use between the foreign agent and the home agent. It also includes the session key to be used between the mobile nodes and foreign agent. The home agent then appends its signature to the resulting message and sends it to the Foreign agent.
6. Upon reception of this message, the foreign agent checks the signature of AAAH, registers the mobile node with the care of address included in the message and stores the two session keys, one to be used with mobile node and the other to be used with foreign agent. It then creates a reply message to the AAAH server, which is digital signed with the static trust relationship between HA and AAAH.
7. The AAAH server then creates a reply message, which includes the reply message sent by the home agent. The resulting message is signed and sent to the AAA server of the visited network.
8. Upon receiving this message, the AAA server of the visited network checks the signature of the message, decrypts, stores, and re-encrypts the session keys using the trust relationship between the foreign agent and the local AAA server. It then sends this message to the foreign agent.
9. Upon reception and successful validation of the signature in the message, the foreign agent concludes that the mobile node is authentic. The foreign agent then decrypts and stores the two session keys to be used with mobile node and the home agent. It then forwards the registration reply message to the mobile node.
10. The mobile node first decrypts the session keys and checks the signature of the home agent. If the check is positive, the mobile node assumes that it has successfully registered at the foreign domain.

After the registration lifetime expiration, if the mobile node wants to re-register, no involvement of the AAA infrastructure is required, as the session keys have already been obtained. When the mobile host moves under a new foreign agent it will attempt
to perform authentication using the obtained session keys. For this, it signs the new foreign agent’s random number with the key it shares with the old foreign agent and indicates the identity of the old foreign agent by including the appropriate NAI extension into its request. The foreign agent then creates an AAA mobile registration request message, which contains the mobile node’s request message and sends it to the local AAA server. The local AAA server then performs a look-up and provides the new foreign agent with the session keys the old foreign agent is holding to communicate with the mobile node and the home agent of the mobile node. Upon reception of this message, the new foreign agent can proceed with the normal Mobile IP registration without the involvement of the AAA infrastructure. The entire procedure involving the AAA infrastructure has to be performed in case of inter-domain handover.

There are some issues associated with this scheme. The authentication procedure is cumbersome as it involves quite a few entities such as the local AAA server and home AAA server, which increases end-to-end latency between the two communicating hosts. Also, the challenge/response verification is distributed. The NAI extension of the mobile node is sent in clear text in the fixed network and this can be a security risk.

3.5.4 Public Key Based Secure Mobile IP

In [Zao+ 99], a design overview of a public key management system called the Mobile IP security (MoIPS) is presented. The MoIPS provides authentication of Mobile IP control messages for location update, access control of Mobile Nodes to resources in foreign networks and secure tunnelling of redirected IP datagrams. To provide these services the authors have proposed a public key based architecture. The registration messages in Basic Mobile IP and the location bindings in route optimised mobile IP carry location bindings. By altering these location bindings or by creating bogus messages or replaying pre-recorded messages an adversary could redirect the traffic generated for one node to another node. In order to circumvent such problems registration and binding update messages must be protected with data integrity, origin authentication and anti replay services. Hence [Zao+ 99] proposes the use of a 64 bit identification tag for detecting replays and one or more authentication tags to provide message integrity and originator authentication.

In order to obtain access control in foreign networks the mobile nodes must complete their registration and attain an attachment point on the visiting subnets. In order to do this the identity and the current status of the mobile node must be verified. In this protocol the identity and network affiliation can be verified by exchanging the public key certificates and demonstrating the possession of private keys corresponding to the public keys in the certificates. On the other hand the status of the Mobile Node can be conducted implicitly by exchanging authenticated registration requests and replies between Foreign Agents and Home agents.

In order for the home network to have the same amount of trust and hence provide the same amount of connectivity to a mobile node when it roams away from its home domain, the home network will require secure traffic tunnelling to and from the mobile node. Similarly in order for the foreign agent to pass traffic for the mobile node the foreign network will require the traffic to be redirected by an authenticated
and trusted entity such as the home agent. These secure tunnels can be implemented by using the IP security protocol in tunnelling mode. Public key based architecture proposed by [Zao+ 99] contain the following three kinds of security support:

1. A scalable key management infrastructure capable of generating and dispatching long term key parameters among any pairs of nodes.
2. A rapid short-term key generation algorithm for supplying short-term keys needed for authenticating the mobile IP and binding update messages.
3. Co-operation of mobile IP and IPSec protocols.

The authors decided to develop a public key infrastructure (PKI) for managing public key certificate and certificate revoking list that are issued to Internet nodes. They also chose to use the Internet Domain Name Systems (DNS) as the primary certificate repository. They chose to use DNS because the Internet nodes are identified by domain names or IP addresses, both of which are carried by DNS. Hence, DNS certificate fetches can easily be piggy-backed to regular exchanges of communication among network entities as this communication is established with DNS lookups. The main advantage PKI based scheme is scalability. Such a scheme has a clear advantage over a distributed system of KDCs. This is because DNS solves the potentially complicated server discovery problem and the use of certified long-term public keys eliminates the need for real time dispatches, as the public keys are issued offline by a certification authority, while in KDCs inline involvement of a trusted third party makes scalability a bottleneck. However, the use of PKI entails a certain amount of overhead.

3.5.5 Secure Mobile IP using Firewall Support

In [Gupta+ 98] a model is proposed to provide mobile users secure access to their company’s firewall protected virtual network. It describes enhancements that enable mobile IP operation by allowing a mobile user, out on a public portion of the Internet, to maintain a secure virtual presence within his firewall-protected office. This constitutes what can be called a Mobile Virtual private Network (MVPN). The architecture for this protocol consists of an interior network and a de-militarised zone. The firewall between the interior network and the de-militarised one is the only point of entry into the organization’s private network. This simplifies the security management.

In [Gupta+ 98], all the mobile nodes belonging to a network as well as those coming from other networks are put in the de-militarised zone. Since the mobile nodes belonging to the corporation have to traverse the firewall to access the private network, they have to authenticate themselves to the firewall using IPSec. Since there is a real end-to-end authentication between the corporation’s own mobile nodes and the firewall, they can easily be configured with secret or public keys. Thus there is one time authentication between the firewall of the home network and the mobile node in the demilitarised zone. The IPSec tunnel provides authentication, integrity and privacy of each IP packet sent during the mobile IP registration procedure over insecure public network. In order to register the mobile node’s care-of-address at the home agent. The message must pass the IPSec tunnel of the home firewall, hence no additional authentication and encryption is required. The objective is that the mobile node continues accessing its home network and its resources exactly as if it were still
within it. It should be noted that when the mobile node leaves its home network, it might migrate both within and outside of the private network's boundaries. As defined by Mobile IP, a mobile node uses a care-of address while roaming.

Until the next movement the mobile node can communicate with any other correspondent node independent whether this is inside or outside the private network. Any data transfer between the mobile node and the correspondent node takes place via the home agent owing to security reasons. It is also possible to communicate with the correspondent nodes directly provided no security is needed. The encrypted and authenticated mobile IP packets are decrypted and decapsulated by the home firewall and delivered to the home agent. The home agent finally decapsulates these Mobile IP packets and delivers them to the appropriate receivers, the correspondent nodes. This approach allows mobile nodes to access firewall protected virtual private networks. This protocol is based on available standards and required minor modifications to the communication stack in the end systems.

This model uses Simple key management for Internet Protocols (SKIP) to secure the traffic. These ideas can also be adapted for the IASKMP/Oakley framework.

The problem with this approach is that its use of a combination of various SKIP and Mobile IP related headers increases the packet sizes and brings about significant reduction in throughput. Also a considerable delay is introduced when the packets have to traverse multiple firewalls since at a minimum, this requires the mobile host to introduce multiple authentication headers, one for each intervening firewall.

3.5.6 Use of IPSec in Mobile IP

The approach in [Zao+ 97] suggests the use of IPSec ESP protocol in the Mobile IP packet redirection tunnels to protect the redirected packets against both passive and active attacks launched and aid these packets to traverse the firewalls surrounding both the home and the foreign subnets visited by the mobile nodes. The purpose is to provide authentication and confidentiality services to Mobile IP redirection traffic and to protect them against passive and active attacks and to help them pass through security gateways. The proposed scheme includes the following [Zao+ 97]:

1. A mechanism for negotiating the use of IPSec protection on selected Mobile IP redirection tunnels.

2. A procedure for establishing these IPSec protected tunnels.

3. The formats of tunnelled packets in either full IP-IP or minimal IP-IP encapsulations.

When implemented on selected packet tunnels, the IPSec data authentication and data confidentiality services enable the mobile node to enjoy the same network connectivity and communication privacy as they were attached to the home network. In this model, the author proposes the use of ESP in tunnel mode. The following assumption are made on the architecture:
1. FAs and HAs providing encryption/decryption and packet filtering based on authentication should be used for the best utilization of this architecture.

2. Firewall protected subnets must enable firewalls nearest to the mobile nodes to function as FAs and all other firewalls in the network should permit the IPSec packet to pass through them.

3. Firewall protected subnets must enable firewalls nearest to the mobile nodes to function as HAs and all other firewalls in the network should permit the IPSec packet to pass through them.

MN-CN, MN-HA, HA-FA and MN-FA are the possible choices of IPSec tunnels. Among these possible IPSec tunnels, MN-CN pair are end-to-end tunnels that may exist regardless of Mobile IP. Among the other three pairs of tunnels, MN-HA tunnels are most useful while MN-FA ones are the least.

FA-HA tunnel: The MIP-IPSec tunnel between HA and FA are the easiest to establish as IPSec protection can be easily added to existing Mobile IP tunnels. When they are used to support data authentication and confidentiality, these tunnels provide a virtual private network connection between home network and foreign network which will see to it that mobile node enjoys the same connectivity even in the foreign network.

MN-HA tunnel: These IPSec tunnels supporting data origin authentication and confidentiality will be the most useful tunnels as they provide a secure communication path between mobile node and home network. The data authentication prevents spoofing and confidentiality frustrates eavesdropping.

MN-FA tunnel: This tunnel is used if no link-layer protection has already been provided. These tunnels provide data confidentiality for a mobile node over the foreign network and data origin authentication for the MN-HA exchange. These tunnels exist only if a mobile node chooses to use a foreign agent care-of address. The establishment of secure channels involves protocols such as IPSec and IKE.

This scheme has the following drawbacks. The MN-HA tunnels are costly to establish, as they are not part of the packet redirection mechanism and always involve a foreign agent’s intervention. Thus causing a bottleneck at the foreign agent and also increasing the number of trusted entities. The MN-FA tunnel are also expensive and hence must be replaced whenever possible. Also every time a mobile host changes its location these tunnels need to be established introducing latency and high costs.

3.5.7 Location Privacy in Mobile IP

Location privacy is very important in cases where the knowledge of the location is as useful as the contents of the transmitted messages themselves. This is considerably important in networks involving military and police. However user anonymity is also becoming increasingly important in business and private environments. Various methods have been described in the architecture and the draft for Mobile IP recommends that if absolute protection from traffic analysis is required then the mobile node must establish a bi-directional tunnel to its HA. In this section we will describe some architecture that provide location privacy features.
3.5.7.1 Security Architectures for Location Privacy

The Mix Method

This method is proposed in [Chaum 81] and is mainly based on public key technology. In this model the sender making the message encrypts the entire message with the public key of the receiving node and then appends his address to the encrypted message. The sending node then encrypts the entire message using the public key of the mix and sends it to the mix. The mix is a trusted third party that collects all the messages from various mobile nodes as a batch and then forwards these messages in a lexicographical sequence. The reliability of the system can be enhanced using a cascade of mixes. All messages are padded with random bit strings to show uniform size. Protection from replay attacks is provided by ensuring that no message is processed more than once.

Though this model provides security from traffic analysis it introduces unacceptable delay as considerable amounts of messages from various nodes have to be collected from various nodes to form a batch. Also the mix is responsible for encrypting the messages to the destination node. This also involves delay. Hence this method is restricted in its applicability to non real-time critical messaging applications. Also the use of a centralized entity causes leads to performance bottlenecks and also reduces scalability.

The Non Disclosure Method:

The scheme proposed by [Fasbender+ 96] is known as the non-disclosure method (NDM). The sender uses the source routing approach to determine which hops the packet must traverse to get to the destination. The sender then encrypts the messages using the public keys of all the hops as specified in the source route header. This way the source node can also ensure that none other than the first node to receive the packet from the sender knows the identity of the sender. Only the last router and the sender know about the receiver node. This way location transparency is achieved.

Unlike the mix method, the attacker here has no chance to find the route. If one of the nodes is compromised, the attacker just gets to know the previous sender and the immediate receiver. However there is no chance for him to locate the next nodes in the path. To be able to sustain location transparency, the source will have to change the path from time to time with out sticking to the same one always. Use of redundancy bits further improves security. The sender first encrypts a message using the public key and then appends this message with the redundancy bits. Then he uses this entire message to compute the new message. The new message which is the combination of the old message and the padded text is again encrypted using the public key of the receiving node. However in this model, the amount of information stored at the node will be tremendously high and this might be a very stringent alternative in the case of mobile applications as the storage capacity at the mobile node is very low in comparison to its static counterpart.
3.6 Summary

The main motive behind this chapter was to introduce the concept of mobile IP and related security issues. One of the most important issues in mobile IP is the notion of location management. In order to communicate with any particular mobile user, it is first necessary to locate the user in the network. This is due to the fact that a mobile user could be anywhere in the area covered by the network. In this chapter, we described some major proposals to providing host mobility and location management in TCP/IP based networks. We have also surveyed some other schemes addressing the location management problem that have surfaced in recent times. Notion of mobility also introduces some problems related to security. Several security parameters are seen by various proposals in the deployment of mobile IP. In this chapter we reviewed some key security proposals that aim to secure mobile IP networks. We also identified a number of drawbacks, which will be addressed in our proposal.
Chapter 4

Distributed Location Management Scheme for Mobile Hosts

4.1 Introduction

In the previous chapter, we looked into a number of schemes for carrying out location management for mobile hosts. In this chapter, we propose a model for location management of mobile hosts in section 4.2. We first discuss definitions and assumptions that govern the foundation of our proposed model. We then describe our model for location management of mobile hosts. We call this model the distributed location management scheme. The proposed model is easy to implement and achieves optimal routing. This model aims to solve some of the problems inherent in the approaches discussed in the previous chapter. In section 4.3, we then discuss tunneling procedures to be used in this framework. In section 4.4, we discuss design issues relating to Transmission Control Protocol (TCP) and Address Resolution Protocol (ARP) support for our model. We then furnish implementation and performance related details in section 4.5. Thereafter, a comparison of these approaches with our model is presented in section 4.6. This is finally followed by a short summary and analysis of our proposed model in section 4.7 followed by some concluding remarks in section 4.8.

4.2 Our Architectural Approach

4.2.1 Definitions and Assumptions

In our architecture, we assume that the system is a collection of autonomous systems and a backbone. The backbone network consists of several routers called Backbone Routers (or BRs) each of which serves a specific region within the backbone known as a cell. The cell helps to exploit frequency reuse and also helps to maintain mobility of hosts, as the bandwidth of mobile systems is limited. A cell might typically correspond to an individual subnet. There is a requirement that the cell be a wireless LAN with wireless 802.11 [IEEE Std. 802.11 99] or WaveLan [Rigge+ 03] support. The BR consists of a transmitter for communication as well as a database for responding to the information processing needs of a mobile host (MH). The total area covered by all the BRs comprises the total coverage area of the backbone. The process of crossing a cell boundary and entering into the other is known as handoff. The MH during the registration phase provides enough information to a BR for the BR to initiate a handoff process with the previous BR of the MH. The actual handoff process is fully transparent to the mobile host and serves to maintain end-to-end data connectivity. A MH can therefore enjoy unrestricted movement within the backbone area. The backbone network itself can support both wireless and wire line environments. The BR acts as a gateway between the wireless cell and the wired network. A MH can move around freely in the backbone area. A MH uses a wireless link layer to communicate with its BR. It is assumed that a mobile host can reside in only one cell at a given instant of time under the jurisdiction of one BR. Where two BRs overlap, a MH communicates with the BR with whom it is currently registered with. On the other hand, a BR can have several mobile hosts in its cell. Please refer to figure 9.
The backbone network is connected to other network systems called autonomous systems (AS) or domains. An autonomous system is a collection of hosts interconnected by a set of routers and subnets under a common and often a single administration. This is referred to as the Autonomous System Controller that acts as an entry point to the AS. In some sense, the Autonomous System Controller acts as a border backbone router. Hence we will use the notation BBR. A BBR serves an autonomous domain whereas a BR does not. A BBR also acts as a location information server for hosts in the AS. It caches the current location information of a mobile host. Any inter-AS communication must pass through the backbone. All traffic generated within an AS to the backbone and from backbone to an AS must pass through the BBR.

The backbone network has a principal location server called the Rendezvous Point (RP). This is a central repository of location information. It maintains a database of mobile hosts that are currently active in the backbone. When a mobile host successfully migrates to a new cell, it becomes the responsibility of its current BR to pass the location information of this host to the RP for updating its database. When a mobile host enters an AS, first it registers with the BBR. The BBR caches this information and sends a copy to the RP for it to update its database. Upon receiving this message the RP realizes that the concerned mobile host is no longer in the backbone area and therefore deletes the relevant entry from its database. The backbone network is further partitioned into zones. The zones as such do not have any administrative significance and are used primarily for fault tolerance purposes as explained later in this section.

A Backbone Router (BR) advertises routes within its cell range using beacon messages. Mobile hosts are always configured to have the BR of the cell act as their default gateway. It performs registration of the mobile host when the mobile host first enters the backbone. It routes packets to and from MHs, tunnelling them to the RP and BBRs if necessary. The BR maintains a database of all mobile hosts currently residing in its cell. It conveys the location information of a mobile host to the RP. The mobile host related information maintained by a BR includes the following: mobile host identity (MH_Id), mobile host entry and expiry times, the multicast groups that a mobile host belongs to as well as mobile host security information. The BBR maintains a database of all mobile hosts currently located in the AS. It performs the registration of a mobile host in its AS and assigns a temporary care of address (CoA) for a certain duration. It maintains a mapping of the permanent and the temporary address of the MH. The information about a mobile host that it stores is similar to that maintained by a BR. The BR, BBR, and the RP must have publicly routable address.

The RP maintains a central repository of all mobile hosts that are roaming in the backbone area (obtained via BRs). It does not include entries for mobile hosts located under BBRs (ie hosts in the autonomous system). It should be noted that the other interface of the BBR points to a backbone cell. For these hosts, the RP must maintain relevant entries in its database as they belong to the backbone. The BBR contains such information as its identity (corresponding to both the interfaces), the mobile host

---

1 Routers within an autonomous system are referred to as Autonomous System Router (ASR), which are capable of supporting mobility.
identity, and security related information. The following multicast groups exist in this architecture:

**All_BBR Group:** This group includes all the BBRs as well as RP.

**BBR_Only Group:** This group includes only the BBRs.

The packet delivery in a mobile IP system such as the one by IETF [RFC 2002] can take place in one of the following two ways:

- Using a care of address
- Using a co-located care-of-address.

In the former case, the care of address is the IP address of some administrative entity responsible for the mobile host such as the foreign agent. In the latter case, the mobile host through some external means in the foreign network acquires the address. Our scheme is based on the former principle. In this case, a BBR or a BR is the end point of the tunnel and on receiving the datagrams decapsulates them and delivers the inner datagram to the mobile host. The advantage of this scheme is that it allows many mobile hosts to share the same care of address and therefore does not place unnecessary demands on the already limited IP v4 address space. The disadvantage of this scheme is that it must always rely on some external administrative entity to support its communications. In our scheme the BRs are widespread and offer mobility related services to mobile hosts such as secure registration and encapsulation. Therefore the processing involved in tunnelling is taken away from the mobile hosts to these mobility support routers thereby minimising the processing at the mobile hosts. The BR and the visiting mobile host have an interface on the same link. In this case, the mobile host and the BR simply bypass their normal IP routing mechanisms when sending datagrams to each other, addressing the underlying link layer packets to their respective link layer addresses. In an AS, there may not be a direct interface between the BBR and the mobile hosts. In which case, we envisage an environment where several local mobility support routers that interface directly with the mobile hosts by acting as a proxy for the BBR. The BBR/BR maintain a binding between the original IP address of the mobile host and the corresponding care of address.

As mentioned before a collection of BRs are grouped together under a zone. Each zone elects one BR to be the coordinator for its zone. This coordinator BR is essentially responsible for keeping track of all hosts within its zone. All such coordinator BRs belong to a multicast group called the All_CBR group. The motivation for this design principle is as follows. In case the RP becomes a bottleneck or succumbs to failure, the nodes in the backbone must be able to still communicate with nodes within the backbone and outside. While no entity but the RP has complete location information, the coordinator BRs can still collectively work together to ensure that all mobile hosts of the backbone remain accessible. One resorts to this search scheme if and only if the RP has encountered a failure. Its use is not recommended in the regular search process. In other words, the coordinator BR acts as a passive location server for its zone as its services are not invoked in the regular operation of the network. A further analysis of failures in the network is given in section 3.7.
4.2.2 Data Structures maintained by DLM Entities

Backbone Router (BR)

A BR maintains both static and a dynamic database. The static database consists of information concerning the BR itself such as its address, and security related information, which are fairly static in nature. The dynamic database contains mobile host related information, which is fairly transient in nature.

Static Data Structure

```
GenInfo {
  BR_id
  All_BBR;
  BBR_Only;
  Broad_Add;
  Mult_list;
  RP_id;
  Cell_Range;
  SecInfo_BR {
    BR_Cert,
    BR_SecKeys & Sec_Pars
  }
}
```

- **BR_id**: Identity of BR.
- **All_BBR**: A multicast group address, which includes all the BBRs as well as the RP.
- **BBR_Only**: A multicast group address, which includes all the BBRs but excludes the RP.
- **Broad_Add**: The broadcast address of the subnet
- **Mult_list**: All multicast groups currently active in its cell.
- **RP_id**: Address of the RP.
- **Cell_Range**: The frequency range in the cell.
- **SecInfo_BR**: Security related information pertaining to BR. (Discussed in the next chapter).

Dynamic Data Structure

```
Struct MHdetails {
  MH_id;
  MH_list;
  MH_Multlist;
  MH_Lifetime {
    Entry_time;
    Expiry_Time
  }
  SecInfo_MH {
    MH_Cert;
  }
}
```
MH__SecRoamProfile

MH Related Information

**MH_id**: The identity of the mobile host  
**MH_list**: List of mobile hosts currently active in its cell. (IP addresses of these hosts).  
**MH_Multlist**: All multicast groups to which a MH belongs.  
**MH_lifetime**: Provided to the MH during registration process indicating its duration of stay in the backbone.  
**SecInfo_MH**: Security related information concerning MH (Discussed in the next chapter.).

Backbone Router Operation

A BR, in general performs the following functions:

1. Advertises routes to the subnet/cell under its jurisdiction using beacon messages. Beacon messages also carry information pertaining to multicast groups currently supported by the BR, and ARP updates for mobile hosts to update their binding caches for some target mobile host.
2. Handshakes, authenticates and registers the mobile hosts visiting its cell.
3. Exchanges state information with other BRs about the location of MHs it needs to serve using handoff procedures.
4. Routes packets to and from its cell to other BRs and BBRs using tunnelling techniques if necessary.

System of Operation: An Overview

A BR periodically broadcasts a beacon message in its cell. For this it must use the broadcast address of the cell (**Broad_Add**). The purpose of this beacon message is to inform a roaming host of its current location and to beckon this host to register with the advertising BR. These messages are broadcast often enough that by missing some of these messages will not cause the MH to assume it is no longer in the cell. The response from the MH to this periodic beacon message is of two types: Registration request message or I am Alive message.

In the former case, the MH generates a registration request (**REG-REQ**) message when it first realises that it has entered a new cell. The MH on receiving such a message realises that it has switched cells and must respond with a **REG-REQ** message. Upon receiving this message the BR initiates an authentication process and thereafter initiates a handoff process with the previous BR of the MH. It then dispatches a location update (**LOC-UP**) message to the RP informing it of the current location of the MH from the registration message; the BR obtains information related to the MH. This includes **MH_id** (MH IP address), and **MH arrival time into the cell** (**Entry_Time**). On the basis of this entry time, the RP calculates the lifetime for the duration of stay for an MH recorded in **MH_lifetime**. Every BR upon greeting a new MH must validate this lifetime field before initiating the authentication process. Upon the expiry of this lifetime, the BR time outs this entry and purges it from its database. Since in the backbone, the hosts are constantly on the move, this lifetime period is
normally small. The other reason to keep the lifetime short is BRs usually handle a large number of mobile hosts. By shortening the lifetime for the roaming hosts appears to be a load-balancing act on the part of BR. The BR also obtains security related information such as a certificate and the keys pertaining to MH that it can use to provide security related services such as confidentiality through encryption. This is recorded in the data structure called SecInfo_MH. A BR also maintains an entry for all mobile hosts that are currently active in its cell in MH_List.

In the latter case, the “I am Alive” message is just to keep the BR informed that it is still serving this MH. This is useful in situations when the MH is not taking part in any active communication session. These messages are Idempotent after the first successful cycle of beacon, registration, registration acknowledgement, and “I am Alive” messages.

To establish communication sessions between mobile hosts, a BR may often have to resort to a search process to locate the target MH. In such cases, the BR has two search options. In case one it simply queries the RP for the location information by dispatching a LOCATE_MH message to the RP. For this reason it maintains RP related information such as the address of the RP (RP_id) and other security related information such as the certificate of the RP along with the necessary keys for secure communication with the RP. If this search process fails, then it uses a multicasting technique to retrieve the necessary location information. For this reason, a BR maintains the address of two multicast groups: BBR_Only group that excludes the RP and ALL_BBR group that includes the RP. In this search process, the BR basically dispatches a WHO-HAS query to BBR_Only group. Once it is able to locate the target MH, then the BR uses appropriate tunnelling techniques to forward the datagrams between the two hosts.

In some cases a mobile host MH may be availing itself services of some multicast group G. A BR must ensure that it caters to this service for a MH by joining the multicast group by attaching itself to the multicast delivery tree. For this reason, the BR must maintain multicast group related information pertaining to each MH such as MH_Multlist (all multicast groups that a MH belongs to) as well as a generic data structure called Mult_List (all multicast groups currently supported by the BR).

**Border Backbone Router (BBR)**

The functions of a BBR are the same as those of a BR. However, in addition to being responsible for mobility management, a BBR is also administratively responsible for its domain known as the autonomous system. As a result of this, it is also fundamentally responsible for vouching for hosts that are on the move but which permanently belong to its AS. In some sense, it acts like a home agent [RFC 2002] for these hosts. Therefore, a mobile host when registering in a foreign AS or the backbone may invoke services of the BBR. In such situations BBR is involved in the authentication process of the mobile host as in the scheme proposed in [RFC 2002]. However, in our case, the search process is on demand using a pull technique rather than a push technique and the packets need not be tunnelled through the BBR as in home agents of [RFC 2002].
Just as a BR, a BBR also maintains two databases: static and dynamic. The static database consists of information concerning the BBR itself such as its identity, and its security related information that is fairly static in nature. The dynamic database contains mobile host related information, which is fairly transient in nature. 

*BBR_id* indexes the static database.

**Struct BBRdetails**

```c
{  
    BBRid;   
    {  
        BBR_idAS  
        BBR_dBB  
    }  
    MH_ListAS;  
    MH_ListBB;  
    Mult_ListAS;  
    Mult_ListBB;  
    Struct Home_Host_List;  
    Home_MHList;  
    RP-id;  
    Struct ACL;  
    Broad_ADD;  
    SecInfo_BBR  
    {  
        BBR_Cert;  
        ALL_BBR_GroupKey & Sec_Par;  
        BBR_Only_GroupKey & Sec_Par;  
        RP_Cert;  
    }  
}
```

*MH_ListAS*: List of MHs residing in the AS  
*MH_ListBB*: List of MHs residing in its cell.  
*MH_MultListAS*: All multicast groups currently active in the AS.  
*MH_MultListBB*: All multicast groups currently active in its cell in the backbone.  
*SecInfo_BBR*: Security related information pertaining to BR. (Discussed in the next chapter).

The dynamic MH related information is maintained as a data structure indexed by MHid.

**Struct MHdetails**

```c
{  
    MHid;  
    MH_Multlist;  
    MH_RoamProfile;  
    MH_Lifetime {  
        Entry_time;  
        Expiry_Time  
    }  
    SecInfo_MH {  
        MH_Cert;  
    }
}
```
BBR Operation

The data structure elements maintained by a BBR are similar to those of a BR. Some notable differences are as follows. A BBR interfaces to both the AS as well as the backbone. A BBR has a unique address on each of these interfaces identified by the elements $BBR_{idAS}$ and $BBR_{idBB}$. Similarly, it is responsible for maintaining location information about the mobile hosts in its backbone cell as well as in the AS. These two types of information are recorded in $MH\_List_{AS}$ and $MH\_List_{BB}$. As mentioned before, a BBR is administratively responsible for all the hosts within its AS. To achieve this objective, it maintains one data structure and an additional element, namely $Struct\_Home\_Host\_List$ and $Home\_MHList$. The data structure $Struct\_Home\_Host\_List$ contains a list of all hosts (both static and mobile) that belong to the AS. In other words they have a permanent IP address assigned to them in this AS. It also contains configuration (not shown) and security related information pertaining to these hosts. Since all traffic generated within an AS to the backbone and from backbone to an AS must pass through the BBR, the BBR effectively acts as a firewall. It is therefore responsible for enforcing access control policies pertaining to its AS. This is maintained using the data structure $Struct\_ACL$.

Rendezvous Point (RP)

The Rendezvous Point (RP) is the principle administrative authority in the backbone network. In some sense it is to the backbone what a BBR is to an AS. It acts as a location server for the backbone. It is responsible for caching the current location information of a MH in the backbone ($MH\_List_{BB}$). It is also responsible for security in the backbone and for initiating security related administrative procedures for authenticating a mobile host from other autonomous systems.

The information held by the RP is as follows:

$MH\_List\_BB$: List of mobile hosts currently roaming in the backbone, their current cell location (Cellid) and governing BR address (BRid). It should be noted that MHS residing in AS under the jurisdiction of a BBR are not recorded in the RP's database.

$MH\_Lifetime$: Provided to the MH during registration process indicating its duration of stay in the backbone.

$MH\_RoamProfile$: Roaming profile of the MH recorded to carry out efficient location searches.

$Struct\_BR\_\{BR\_Config\_Info\}$

Rendezvous Point Operation

Any mobile host when first entering into the backbone must authenticate itself to the RP. The RP is then responsible for location management functions with respect to this mobile host. The RP is also responsible for initialisation, configuration and management of BRs in the backbone (using the data structure $Struct\_BR$). It assigns every MH that has entered into the backbone a lifetime that determines the duration of
its stay in the backbone (MH-Lifetime). At each BR, this lifetime is validated and upon its expiration the MH must re-initiate its registration process with the RP. The MH also has the option of renewing its lifetime after \(3/4\) of its lifetime has expired.

**Mobile Host (MH)**

The mobile host (MH) is a host that can freely move and yet at the same time maintain seamless network connectivity with the communication infrastructure. The mobile host is primarily responsible for maintaining information pertaining to it.

A MH also maintains both a static and a dynamic database. The static portion contains details pertaining to the MH whereas the dynamic portion consists of information pertaining to the MH's location. The static MH related information is maintained as a data structure indexed by the Mhid.

**Struct MHdetails**

```
struct MHdetails{
    Mhid;
    MHmultlist
    SecInfoMH
        {
            MH_Cert;
            MH_Sessionkeys & Sec_Par;
            MH_Groupkeys & Secpar;
            MH_RoamProfile;
        }
    }
```

The information elements are as described above.

The dynamic portion of the database is again divided into two sections: One pertaining to the backbone and the other pertaining to current cell location. The backbone section remains static for the duration of stay of MH in the backbone. On the other hand, the information pertaining to current cell location is fairly transient as it gets updated every time MH switches its cells.

Information pertaining to the backbone is maintained as data structure indexed by the RPid.

**Struct Backbonedetails**

```
struct Backbonedetails
    {
        RP_id;
        RP_Cert;
        MH_Lifetime {
            Entry_time;
            Expiry time;
        }
    }
```

Information pertaining to current cell location is maintained as a data structure indexed by the current BRid.
The information elements are as described above.

Mobile Host Operation

The mobile host MH must be capable of determining a change in its location and then be able to initiate appropriate registration procedures. To achieve this, it must constantly monitor the beacon messages of the BR/BBR and respond to these messages by initiating the registration procedure. It must carry its credentials to support its operational and communication related functions on the move (as in the data structure Struct MH). In its credentials, it also maintains information about its home AS such as BBR_id and the certificate of the BBR. It must also maintain information pertaining to the backbone for the duration of its stay in the backbone. Some critical elements are the identity of the RP (RP_id), the certificate of the RP along with its lifetime details. It also needs the details of the BR in its current cell location. For instance it retrieves the certificate of the BR from the beacon message and initiates a secure registration process using this certificate.

4.2.3 Distributed Location Management Scheme (DLM)

A mobile host (MH) is assumed to be assigned a permanent address called MH_id. This identity may be assigned to the host by its organisation or through some other independent scheme (such as through an Internet Service Provider). The mobile host retains this identity in the backbone. At each point of attachment, the mobile host functions under the care of address of a BR. All packets to and from the MH must be encapsulated using some IP in IP encapsulation technique. We suggest the use of generic record encapsulation (GRE) for reasons explained below. It is necessary for the BBR to support such a tunneling service.

Let us first consider the mobile router discovery process. A router that supports mobility (BR, BBR) periodically broadcasts a beacon message. In other words, the BR/BBR advertises its services on a link. By means of these advertisements, a host can determine if it has changed its current cell location or moved from the backbone to an autonomous system region. Apart from this, the beacon message can also be used for the following purposes. A BR/BBR can use these messages to advertise what multicast groups they are currently serving. If the mobile host belongs to one of these groups then it can begin to receive the datagram for this group immediately rather than querying the BR and incurring a delay. These beacon messages can also be used to update the address resolution protocol (ARP) caches of mobile hosts in the cell. A detailed description of ARP resolution and related issues is explained in section 2.5.1.

Assume that a MH enters an autonomous system region. Upon receiving a beacon message from a backbone router, the MH realises that it is no longer in the AS region. The MH then responds to this beacon message by sending out a registration request \((REG\_REQ)\) that includes its address and profile as explained in the section on data structures. The BR upon receiving this \(REG\_REQ\) message passes it on to the RP.
RP authenticates and registers the mobile host and caches MH related information in its database. If the authentication process is successful, the RP responds with a Registration Accept (REG_ACCPT) message otherwise it responds with a registration denied message (REG_DENY). The MH also gets a token from the BR that contains amongst others the following important details:

- How long is the MH registered in the backbone: The duration of stay of the MH in the backbone. This is typically also the lifetime of the token.
- Security related details: This reveals that the mobile host has been successfully registered with the RP. Other BRs need not perform the authentication process again.

However, once the lifetime of the token expires, the MH must initiate a fresh registration process. The initial steps that the mobile host undergoes to receive the token when it first enters the backbone is called the registration phase. Thereafter, when the mobile host begins to move in the backbone and switches the cell it responds to the beacon messages of the BR with a greeting message (GREET). The greeting message amongst other things simply contains the token of RP that a MH must use to authenticate itself to the BR. The BR validates the token (actual process of validation is explained in the next chapter) and proceeds to offer the necessary services for the mobile host.

Every time the mobile host switches its cell a handoff process occurs. Using the GREET message the mobile host conveys the identity of the previous BR responsible for this host. The current BR upon validating the host then initiates a secure handoff process with the previous BR on behalf of the mobile host. After the successful completion of the handoff process, the current BR generates a location update (LOC_UP) message for the RP. The RP responds with an acknowledgement message (LOC-UP_ACK). The previous BR deletes the entry for this mobile host from its database. Since the MH is roaming, it moves from one cell to another, each of which is under a different BR. The only two entities, which need to be aware of the current location of an MH, are its current BR and the RP.

When a MH receives a beacon message from a BBR, it realises that it is within an AS. It sends a REG_REQ to the BBR and supplies its original MHid along with the lifetime desired. The BBR either responds with a REG_ACCPT message or REG_DENY message. Upon successful completion of the registration process, the MH comes under the Care of Address (CoA) of the BBR. The BBR maintains a database that essentially contains a list of MHs that are currently residing in the AS. BBR also generates a LOC_UP for the RP. The RP upon receiving this message realises that the mobile host is now longer in the backbone area. The RP goes through its database to locate the entry for this host and purges it from the database.

To reduce the location search cost, the MH user’s profile can be built by monitoring his/her activities. Consider this scenario. An executive may connect to his office network in the morning from home. While on the way to work, he continues to interact with his office network via the services of BRs. Finally when he reaches his office (some AS) his connection is then maintained by the BBR. The route taken by the executive to reach his office daily is a fairly static one. The time duration can also be predicted before hand fairly accurately with slight variation. Such a profile of a
mobile host can be cached by the RP as well as the BBR of the mobile host. The packets can be tunneled directly to the current location of the mobile host without having to undertake a search process. Other hosts with the help of the RP and the BBR can retrieve this profile of the MH after the necessary security verification. This information is cached by the RP in the data structure `MH_RoamProfile`. This knowledge of the profile of the MH can also help in the provision of Quality of Service related services for mobile hosts as the movement and needs of the MH can be predicted beforehand.

For the management of the network and to carry out the search process the system maintains several multicast groups and messages. All the BBRs including the RP form a multicast group called `ALL_BBR` group. When a BBR acting on behalf of a sender’s query wants to determine the current location of a target mobile host it multicasts a message `WHO_HAS` to the `ALL_BBR` group. The BBR/BR or the RP that is currently responsible for this MH responds with an `I-HAVE` query. A BR when searching for a target mobile host MH generates a location query message `LOCATE_MH` for the RP. If the RP can track down the BR that is responsible for the MH then it responds with these details in the `LOC_INFO_MH` message. Otherwise, it responds with a negative acknowledgement message `NEG_MESG`. In response to a
NEG-MESG message the BR generates a multicast query WHO_HAS to BBR_Only multicast group. This group does not include the RP. The concerned BBR will then respond with an I-HAVE message. In case the RP fails, the network initiates a fault tolerance mechanism by means of which all the coordinator BRs collectively cooperate to provide the ongoing communication service. The coordinated BRs form a multicast group called ALL_CBR group whose services are invoked to carry out location management and in general the network operation functions.

Let us now consider the various scenarios whereby a mobile host moves between autonomous domains and the backbone. We will first consider five cases and then put them together and describe the overall behaviour of each of the components involved in our distributed location management architecture. We will refer to the host that the mobile host is trying to communicate with as the Correspondent Host (CH).

**Case 1: Mobile Host and the Correspondent Host in the same AS**

Assume both the mobile host MH1 and the correspondent host CH are currently registered in an autonomous system AS1. It is assumed that MH1 has already undergone the process of registering with the autonomous system controller BBR1. Assume that CH wishes to communicate with MH1. CH generates packets for MH1, which are received by its local router (some ASR) and then forwarded to BBR1. BBR1 looks at its visitor's directory to locate MH1. If BBR1 finds MH1, it tunnels the packets to MH1’s current location using its care of address. Thereafter, MH1 can communicate directly with CH.

**Case 2: Mobile Host and the Correspondent Host are in different AS.**

Assume that a mobile host MH1 is in an autonomous system AS1 and the correspondent host CH is in an autonomous system AS4. CH wishes to communicate with MH1. It is assumed that MH1 has registered with the autonomous system controller BBR1. BBR1 updates its domain list entry as well as RP. Packets generated by CH are received by its BBR4. BBR4 checks its registration list to verify if MH1 is located in AS4. If MH1 is not located in AS4, BBR4 dispatches a WHO_HAS query to All_BBR group. Note that in this case RP will not contain location information of the MH as it is residing in an AS. Upon receiving this query message, BBR1 checks its database and finds the concerned MH to be under its jurisdiction. It sends back to BBR4 an I-HAVE reply. From BBR1’s reply, BBR4 can find out that the MH is in BBR1’s jurisdiction. Now BBR4 can forward the packets to BBR1, which in turn sends them to MH1.

**Case 3: Mobile Host in the Backbone and the Correspondent Host in an AS**

---

2 Due to a delay in the network, the RP may not be updated on time. In such cases, a query may receive two positive answers, one from the BBR and one from the RP. The assumption here is that as an MH is intended to be mobile and its stay within an area may be for a short time, the information from a BBR can be more up to date than the reply from the RP.
Assume a mobile host MH1 is in the backbone currently under the jurisdiction of BR1. We will assume that MH1 has already undergone the process of registering with the RP via BR1. Assume that a CH is in an autonomous system AS1 and wishes to communicate with MH1. The packets generated by CH are forwarded to its BBR1. BBR1 checks its registration list to verify if MH1 is currently in its domain. If not found, as it will not be in this case, BBR1 dispatches a WHO_HAS query to the All_BBR group, which includes the RP. RP checks its domain list entry and finds the concerned MH1 under BR1 in the backbone. It sends back to BBR1 an I-HAVE reply. Then BBR1 can forward the packets to BR1, which then forwards them to MH1. However, if we assume that MHs change their location quite frequently within the backbone and as a BR does not cache the location information of MH after a hand-off, then it may be more appropriate for BBR1 to send the packets to the RP, which can then transfer to the right BR and MH1. On the other hand this approach could lead to the RP being a bottleneck.

**Case 4: Mobile Host and the Correspondent Host in the Backbone**

Assume that both the mobile host MH1 and the Correspondent Host CH are residing in the backbone under the jurisdiction of routers BR1 and BR2 respectively. It is assumed that both MH1 and CH have already undergone the process of registering with the RP via their respective mobile routers. Let us assume that CH wishes to communicate with MH1. CH generates packets for MH1 and forwards them to BR2. BR2 checks its local cache to find out if MH1 is currently registered. If so, the packets are forwarded to MH1. If not, as it will not be in this case, BR2 sends a LOCATE_MH query to RP. Once again we have the two options mentioned above. RP can respond with an I-HAVE reply in which case BR2 forwards the packets to RP and then RP forwards them to BR1. Alternatively the RP can send to BR2 the information that MH1 is under BR1 and then BR2 forwards the packets to BR1. The same issues discussed in Case 3 above apply here.

If MH1 is under a BBR (if the host in the AS) then the RP will not know of its location. In such a case, the following search process occurs. BR2 sends a WHO_HAS query to the RP. The RP sends back a negative acknowledgment to BR. BR then sends a Who_Has query to BBR_Only group. It should be noted that this group does not include the RP. The concerned BBR responds with an I-HAVE message. All the packets are then subsequently routed to BBR from where they can get delivered to the intended recipient MH1.

**Case 5: Mobile Host in the AS and the Correspondent Host in the Backbone**

Assume that the mobile host MH1 is in an autonomous system AS1 and CH is in the backbone. Let us again consider the case where CH wishes to communicate with MH1. The earlier part of the process is same as the one described in Case 4 above. BR2 finds out that MH1 is under BBR1. Now BR2 can forward the packets to BBR1, which in turn sends them to MH1.

The pseudo code for the cases presented above is as follows:

```
If sender and the target MH are in the same AS
    deliver/route using regular IP code.
else if sender and target MH are in different ASs
    
```
search (All_BBR group) to locate MH, deliver to the corresponding BBR;
else if MH is in the backbone and sender is in AS
search (All_BBR group) to locate MH, deliver to the RP;
else if sender and mobile host are in backbone
search (by querying RP), deliver to the RP;
else if sender is in backbone and target MH is in AS
Search (by querying RP), (BBR_Only Group) to locate the MH,
deliver to the corresponding BBR;

4.3 Tunnelling Procedures

A tunnel is a path followed by the datagram while it is encapsulated. The conceptual model is such that, while it is encapsulated, a datagram is protected from normal Internet routing until it reaches a knowledgeable decapsulating agent. In other words, tunnelling is a process of bypassing the normal Internet routing of a packet by enclosing or encapsulating the packet within a new IP header containing an alternate destination IP address [Perkins 98]. There are various encapsulation techniques available for use with IP. The principle encapsulation techniques are discussed below.

4.3.1 IP-In-IP Encapsulation

This encapsulation technique is straightforward. An outer IP header is added before the original IP header. The outer IP header source and destination addresses identify the endpoints of the tunnel. The inner IP header source and destination addresses identify the original sender and recipient of the datagram. Between the outer and inner IP headers, other additional headers such as security headers (e.g. authentication headers) can be added that are specific to the tunnel.

In this tunnelling scheme, the inner IP header is left untouched for the duration of the tunnel. At the encapsulating end, the encapsulator decreases the TTL value of the inner header by 1 if tunnelling is being done as part of forwarding the datagram. Otherwise the inner IP header TTL is not changed during encapsulation. The TTL of the inner packet then remains the same until the tunnel exist point where the depasulator decrements the TTL value by 1. Thus as far as the inner packet is concerned, the tunnel entry and exit points are just one hop away. Other hops within the tunnel remain transparent to the inner IP packet. It must be noted that the security options, if any, of the inner IP header may affect the choice of security options for the encapsulating outer IP header [Perkins 98]. Encapsulator will not tunnel a datagram that has a TTL value of 0. It must discard such packets and generate appropriate ICMP message to inform the sender of the same. Similarly, the decapsulator will also discard any datagram that has a TTL value of 0.

While IP-In-IP encapsulation appears to be a simple and elegant way of tunnelling datagrams it introduces an unnecessary duplication of several fields within the inner IP header causing processing overheads.

4.3.2 Minimal Encapsulation

This is a scheme in which an IP datagram may be encapsulated (carried as payload) within an IP datagram, with less overhead than IP-In-IP encapsulation that as
previously mentioned, adds a second IP header to each encapsulated datagram. The IP
header of the original datagram is modified and the minimal forwarding header is
inserted into the datagram after the IP header, followed by the unmodified IP payload
of the original datagram that contains the transport header and transport data. In
encapsulating the datagram, the original IP header is modified as follows [RFC 2004]:

1. The protocol number 55, for the minimal encapsulation protocol, replaces the
   protocol field of the IP header.
2. The destination address field in the IP header is replaced by the IP address of the
   exit point of the tunnel.
3. If the encapsulator is not the original source of the datagram, the source address
   field in the IP header is replaced by the IP address of the encapsulator.
4. The total length field in the IP header is incremented by the size of the minimal
   forwarding header added to the datagram. This incremental size is either 12 or 8
   octets, depending on whether or not the original source address present (S) bit in
   the forwarding header. The S bit signifies if the original source address is present
   in the datagram.
5. The header checksum field in the IP header is recomputed or updated to account
   for the changes in the IP header.

Note that unlike IP-in-IP encapsulation [RFC 2004], the Time to Live (TTL) field in
the IP header is not modified during encapsulation; if the encapsulator is forwarding
the datagram, it will decrement the TTL as a result of doing normal IP forwarding.
Also, since the original TTL remains in the IP header after encapsulation, hops taken
by the datagram within the tunnel are visible, for example, to "traceroute". Minimal
Encapsulation has 8 or 12 less bytes of overhead than does IP in IP Encapsulation.
However, this scheme is only applicable for unfragmented packets as there is no space
left for fragment identification.

4.3.3 Generic Record Encapsulation (GRE)

This encapsulation technique is similar to IP-In-IP encapsulation but more generic in
nature. It provides features by means of which IP can be used either as a delivery
protocol or as the delivery and payload. In the latter case, the TTL, Type of Service
(TOS), and IP security options may be copied from the payload packet into the same
fields as the delivery packet. The payload packet’s TTL is required to be decremented
when the packet is decapsulated to ensure that the packet cannot be forwarded
indefinitely.

The advantage of GRE over previous two methods of encapsulation is that it can
encapsulate numerous other protocols besides IP. However, it does not minimise the
overhead introduced due to duplication of fields resulting in large packet sizes.

4.3.4 Remarks

Certain requirements may govern the choice of which encapsulation technique to use.
In our model security was a key consideration and therefore IP-In-IP and GRE was
preferred over minimal encapsulation. This is because there may be security
requirements to protect the original payload during tunnelling. GRE offers additional
security through strict source routing option. For our scheme, we are trying to provide
facilities for both IP-In-IP and GRE techniques. Details are furnished in the section on implementation.

4.4 Other Design Issues

There are some other design factors that a location management scheme must take into consideration to offer seamless connectivity service for the mobile hosts. Two critical questions that need to be addressed are as follows:

- How is the IP address to hardware address resolution handled for a roaming host?
- What challenges does mobility present for higher protocols such as the transport protocols and how do we resolve these issues?

We will try to address these issues in this section.

4.4.1 Support for Address Resolution Protocol (ARP)

The use of ARP [RFC 826] requires special rules for correct operation when mobile nodes are involved [Perkins 98]. The requirements specified for ARP in this section are similar to those specified in [RFC 2002] and [Ioannides+ 93]. Our model offers the capability to support proxy ARP and gratuitous ARP. However due to some inherent deficiencies associated with these schemes, we have optionally designed a better mechanism for propagation of ARP messages that we recommend be used. We shall first explain the regular schemes and then suggest our recommended solution.

A proxy ARP [RFC 925] is an ARP reply sent by one node on behalf of another node that is either unable or unwilling to answer its own ARP requests. When the mobile host MH leaves its own autonomous system, the BBR for that autonomous system uses proxy ARP as follows. If a packet arrives at the BBR destined for the mobile host, the BBR has the following options. It can initiate a search process to determine the current location of the mobile host and then tunnel the packet to the current location of the mobile host. Alternatively, it informs the sender of the packet with the current location details of the mobile host. If the sender of the packet belongs to the same autonomous system as the MH then BBR opts for the tunnelling option. In this case, the BBR will proxy ARP for this target MH. This implies that the sending host will associate MH’s IP address with the BBR’s MAC address. When this happens the sender will send all traffic through the BBR, which uses IP encapsulation techniques mentioned above to deliver the datagram to the target MH. If the sender does not reside in the autonomous system it may well be the case that the datagram may not arrive at this autonomous system in the first place, as the sender would have initiated a search at its end as explained above. However, some datagrams may still arrive at the home autonomous system. In such situations we recommend that the BBR provide the sender with the current location details of MH. The pseudo code for the ARP reply implemented in the BBR is as follows:

If target is not an MH then
    Use the regular ARP code;
else if source is a remote/unknown MH
    Initiate a search for target MH, provide the sender with location details;
else if both source and target hosts are local
Drop the packet;
else if source is local and target is remote then
Initiate a search for target MH, proxy-ARP for target;

In gratuitous ARP [RFC 2001], an ARP packet is sent by a node to update other nodes’ ARP caches. In our model, when a mobile host MH belong to an AS is roaming and the BBR has to broadcast an ARP reply indicating that MH’s MAC address is BBR-MAC (BBR’s MAC address). Until MH returns home to its AS, the BBR will continue to proxy ARP for MH. When MH returns to its home AS, all nodes on the MH’s home network must once again learn to associate the MH’s own link layer address with the MH’s IP address. The MH and the BBR use gratuitous ARP for this purpose.

Gratuitous ARP is inherently an unreliable mechanism as it depends on error free reception of the gratuitous ARP messages. Also this scheme may generate too many messages depending upon how frequently a mobile host moves. As an option, we recommend the use of regular beacon messages to flush out the ARP caches of all the MHs registered with the BR/BBR when a MH migrates. This minimises the number of control messages and enhances reliability.

4.4.2 Transport Protocol Operation over DLM

Our model, in general, provides transparent network access to every entity above the network layer ensuring seamless and transparent mobility. However, there may be minor hiccups experienced at the transport layer especially when running TCP over DLM. The problem stems the fact that TCP makes an implicit assumption that packet losses are largely due to congestion. This assumption does not hold true in the mobile environment. In mobile networks, TCP connections encounter types of delay and loss that are unrelated to congestion [Caceres+ 94]. For example, with respect to the DLM model, communication may pause while the handoff between BRs complete and then routing of packets between mobile hosts resume. Second, packets may be lost due to futile transmissions over the wireless network when a mobile host moves out of reach of other transceivers, especially in networks such as ours where we assume that there is little or no overlap between cells in the backbone. Third, all transmissions between mobile host and fixed infrastructure occurs over wireless links. Since these links are inherently unreliable, packets may be lost due to relatively frequent transmission errors. Some performance degradation due to these delays and losses is unavoidable. However, since TCP is unaware of these events, it only triggers standard congestion control procedures that further degrade network performance.

Several schemes have been suggested to improve performance of TCP over wireless links. These schemes can be classified into two classes. In the first approach, the TCP sender is unaware of the losses due to the wireless links so the TCP at the sender need not be changed [Bakre+ 95, Bala+ 97, Brown+ 97]. In the second approach, the sender is aware of the existence of the wireless link in the network and attempts to distinguish the losses due to wireless link from those due to congestion. So the sender does not invoke congestion control algorithms when the data loss is due to wireless link [Caceras+ 95, Stangel+ 98, Biaz+ 97].

83
While our model attempts to make host mobility transparent to higher layer protocols such as TCP, from the above discussion it can be argued that these protocols may actually benefit from the knowledge of mobility. Although this is an issue that is beyond the scope of this dissertation, we would still like to make one final observation. Because the DLM model consists of both mobile and stationary hosts, it becomes logical to support the option where no changes are required at the sender. It is reasonable to make a case such as this as the sending hosts (especially the stationary hosts) need no modification to the TCP stack.

4.5 Implementation

Our implementation environment consists of two autonomous systems (Refer to figure 10). Each AS network consists of a wired part and a wireless part. Two routers WR1 and WR2 integrate these parts – with one wired Ethernet interface and one wireless Ethernet interface which act as IP forwarders. The BBR module is being implemented in a Pentium 120 MHz with 32 MB RAM. Attached to the wireless side of this network we have the mobile host Toshiba Satellite and one wireless Lucent WaveLAN bonze interfaces. The mobile host communicates with the BBR via WR1 and one other router R that separates this network from the network where BBR resides.

All wired interfaces used have a peak bandwidth of 10 Mbps, whereas the wireless equivalencies are 2 Mbps. From the software perspective are codes are being compiled on Linux 2.2.4 kernel according to our network conditions. To accommodate mobility, this Linux kernel has to be recompiled to support features such as IP in IP tunnelling. This compilation results in the kernel module ipip.o. Following a few basic configuration commands, the following sample script does the job. (For further details refer to [IP Tunnel 02]).

```bash
start()
{
    modprobe ipip
    iptunnel add ${TUNDEV} mode ipip remote ${REMOTEIP}
    ifconfig ${TUNDEV} ${GATEWAY} pointopoint ${REMOTEPRIVATEIP}
    route add -net ${REMOTENET} gw ${GATEWAY} dev ${TUNDEV}
}
```
stop() {
    iptunnel del ${TUNDEV}
    modprobe -r ipip
}

case "$1" in
    start)
        start ;;
    stop)
        stop ;;
    restart)
        stop
        start ;;
    esac

This script results in an IP-In IP tunnel. We intend to perform GRE using GRE pseudo interfaces, which simulate point-to-point connections. Each such GRE interface enabled, by default, supports two modes of operation: GRE encapsulation (default and Minimal encapsulation. We intend to use GRE encapsulation. It is also the default mode on all Cisco routers.
4.6 Comparison with other prominent approaches

In this section we compare four prominent mobile IP proposals with our approach. The criteria for comparison have been identified in [Myles+ 93]. In [Myles+ 93] a solid groundwork is done for the evaluation of four prominent mobile IP protocols. We extend this evaluation model by including the IETF mobile IP scheme and our scheme. The schemes to be evaluated are:

- Columbia Mobile IP. [Ioannidis 92a, b, c, 93a]
- IBM Mobile IP. [Perkins+ 94]
- Sony Mobile IP. [Teroaka+ 92]
- IETF Mobile IP. [RFC 2002]
- Distributed Location management approach (Our model)

We take the same set of criteria to assess our model. These protocols have been chosen as they are popular standards with complete specifications. As cited in the previous chapter, there are other proposals too. Since many of these other proposals use very similar techniques and have many features in common with these mobile IP protocols discussed we chose to ignore them.

In [Myles+ 93], the following factors are taken into consideration when evaluating a mobile IP protocol:

- **Backward Compatibility**: Any protocol that is prosed for mobility must be compatible with existing network protocols. The authors argue that the investment in the existing network infrastructure is too large to discard simply for the sake of mobility.
- **Optimum Routing**: For any mobile IP protocol to be successful, it must be capable of transporting packets through the network fabric using optimum or close to optimum routes. In [Myles+ 93] an optimum route is defined as the route that existing routing protocols would use between two fixed hosts. Sub-optimal routing results in wasted network bandwidth, extra processing and a lower overall level of performance delivered to the user. In terms of mobility requirements, it will impact the performance transparency to the user.
- **Intra-cell Communications**: This implies that two hosts that are connected to the same data link medium should be able to communicate directly without the assistance of any third party such as a base station or a mobile support router. Depending on the lower level protocols used, direct intra-cell communications may halve bandwidth requirements.
- **Migration Procedures**: The migration procedures must be efficient and reliable so that communications can restart as soon as possible after a MH migrates to a new location. Long migration procedures will result in unacceptable breaks in network connectivity.
- **Packet Loss**: The network must minimise packet losses. There should also be efficient recovery mechanisms in place when a packet is lost. This may require the protocol to generate additional control messages for management purposes. A trade off is between degrees of losses to the overhead of the control messages.
• **Component Loss**: What this means is that a mobile IP based protocol must be able to survive the loss of major components. The recovery probably does not need to be very fast, as major components do not fail very often.

• **Network Breakage**: Any ideal mobile IP based protocol must facilitate continued communication service by means of which a mobile host can seamlessly communicate with other hosts on the same segment, even if the network is broken. Network breakage is similar to the loss of a large number of components and generally the same techniques can be used to increase robustness.

• **Processing bandwidth**: A mobile IP protocol that requires large amount of processing will have lower network performance or higher cost. Lower network performance will affect performance transparency.

• **Impact on existing routing protocols**: A mobile IP protocol must not place an additional load on existing protocols such as RIP and OSPF.

• **Address Usage**: A limitation of IPv4 is that it has almost run out of address space. It is thus desirable that a mobile IP protocol does not put additional pressure on this scarce resource.

• **Security**: Security requirements are often not well understood even in the context of non-mobile hosts. The use of mobility enhances the complexity associated with this problem.

Let us now evaluate the existing proposals against this set of criteria.

1. **Backward Compatibility**: The IBM mobile IP scheme is based on the use of Loose Source Routing (LSRR) a facility not often used in the past. Although most routers process LSRRs correctly, it has been demonstrated by [Ioannidis 92a] that very few existing hosts implement [Braden 89]. In fact most existing TCP/IP implementations upon receiving a packet with a LSRR option simply return it as a recorded route. Those few implementations that correctly reverse the LSRR, as required by [Braden 89], do not track changes in the LSRR during a connection. Instead they always use the LSRR specified at the connection set up. An even greater problem occurs with UDP. In [Braden 89] it is stated that UDP must pass received LSRRs to the application, which is then supposed to reverse the LSRR for the reply. This may cause problem. For instance, Unix sockets do not even have the ability to pass the appropriate information across the Application Programming Interface (API) and no known UDP application performs the correct LSRR reversal. Correct operation of the IBM proposal would therefore require changes to applications, which is unacceptable in terms of backward compatibility.

The Sony Mobile IP scheme also makes use of the options field. Some existing implementations drop any packets with options [Myles+ 93]. Until such communications are fixed communication using Sony or IBM schemes will always have some problems. In contrast Columbia mobile IP and IETF mobile IP schemes have no specific backward compatibility problems. Our model does not restrict routing to any particular option feature of IP. Hence such problems do not surface in our approach.

Another issue is the support for multicast. Sony, Columbia, and IBM schemes do not have support for multicast routing protocols. Hence a group-based communication cannot be carried out in an optimised way in such schemes. IETF
mobile IP scheme does provide the necessary infrastructure. However this scheme does not achieve optimal multicast routing. In some situations, when the sender and the receiver are on the same network must still traverse the entire internetwork twice. This scheme also suffers from tunnel convergence problem resulting in packet duplication in a grand scale causing enormous overheads.

In contrast our approach uses existing multicast infrastructure in the backbone (PIM- SM) to facilitate multicast communication. It requires no changes to the multicast protocol and scales well for the large internetwork. It also achieves optimal routing as the packets are not forced to get routed via the home agent of the home domain of the roaming mobile host.

2. **Optimum Routing:** In Sony Mobile IP scheme, packets get routed in an optimum fashion as each MH caches the current location of the other MH. However in some situations a problem may arise. Consider this situation as described in [Myles+ 93]. In this case, a fixed host may send packet to a mobile host and there are very few Sony routers in the topology. All the packets from the fixed host to the MH will be routed sub-optimally via the home gateway.

In the IBM scheme, always close to optimum routes are obtained if both end points of the path implement LSSR correctly. However, as mentioned before, most existing host implementations do not process LSSRs correctly so this scheme will, in the worst case, route all packets destined to a mobile host sub optimally via its home Mobile Router.

The Columbia also uses close to optimum routes when operating in the local area. In IBM and Columbia schemes, a small number of extra hops will be taken depending upon where the BASs and MSRs are placed within the network topology. However, in the wide area, the Columbia scheme uses sub optimum routes when a MH in the popup mode communicates with a host not attached to the MH’s home network. In such cases, the route between the two MHs is through one or two MSRs connected at the home network.

In the IETF mobile IP scheme, the packets are subjected to triangle routing. Packets from a sender to a mobile host are routed to the home agent of the mobile host. The home agent using appropriate tunnelling techniques then delivers the packet to the current location of the mobile host. Hence less than optimised routes are achieved using this technique. However there is an extension to this approach known as route optimisation. Route optimisation allows a source to cache the binding information and tunnel packets directly to the mobile host. The problem of how to maintain cache consistency, who should update the cache entries and on what basis should the updates be propagated are still being investigated [Badrinath+ 96]

Our scheme offers the best approach to optimum routing in all situations. Any fixed host initiates a search locally using its home router (BBR) to obtain the current location of the target mobile host. A maximum of two search techniques will yield the current location of the mobile host. The packets are then routed in most optimum fashion to the target mobile host without the intervention of the home router or the home network of the mobile host. Since for any query first a
local search is initiated, we avoid the triangle routing problem that occurs in Columbia scheme when the host is in the popup mode.

3. **Intra Cell Communications**: In IBM scheme, short ARP timeouts are suggested causing too many unnecessary ARP timeouts. Longer time outs will result in noticeable breaks in network service to the user.

In Sony a fixed host and a MH in the same subnet can take part in direct intra-cell communications. However when the MH migrates the approach resorts to the use of gratuitous ARPs or ARP timeouts. This is the same as the approach used in Columbia mobile IP and IETF scheme.

In our approach we provide the support for gratuitous ARP schemes. However we embed ARP updates in the regular beacon messages to make it more reliable and fast. The down side of this scheme is it generates traffic over heads, which is proportional to the rate at which the MHs migrate out of the network.

4. **Migration Procedures**: In the Sony mobile IP scheme the migration procedures are inefficient. Consider a situation cited in [Myles+ 93]. If two mobile hosts are communicating and if one of them migrates, this host must notify its home gateway, which will then attempt to delete all cache entries throughout the network for the migrating MH. The scheme uses VipDelAamt packets for this purpose. Therefore all routers along the way must have an understanding of the Sony protocol. According to [Myles+ 93] this is unlikely to be the case in the long term and so the cache deletion procedure will invariably fail. This scheme also potentially makes inefficient use of network transmission bandwidth. Every time a MH migrates to a new location an attempt is made to update all routers and hosts that contain information about the MH’s location. This update process can use a large amount of bandwidth.

In contrast, the Columbia and IBM schemes update locate information at one or two places. Other updates are done on an as-needed basis. In IETF mobile IP scheme the mobile host only needs to register its current location to its home agent every time it changes its location. In the original scheme the moving host does not inform its new location to its previous foreign agent. This will either result in a packet loss or high latency for delivery, as packets might have to be again tunnelled to the current home agent and then back to the current foreign agent.

In our approach these problems are mitigated to a great extent. Our approach does not force location update messages to be propagated to the home network every time a mobile host MH changes its location. For instance, in the backbone, an MH may need to register just once with the home agent. The RP and the current BR maintain the location information. Every time a handoff occurs, the new BR and RP update their caches with the MH details while the previous RP deletes its cache. At any given time, only a maximum of two entities maintain location information making this scheme efficient. Also it does not introduce a single point of failure as in IETF where a home agent failure may make all the hosts it is responsible for inaccessible. This is because a search process does not necessarily depend on a MH’s home BBR to provide the current location information for the mobile host.
5. **Packet Losses**: IBM and Columbia mobile IP schemes use acknowledgements to ensure that important management packets are received correctly. If an error does occur then the recovery occurs after one or more retransmissions.

However, in the case of Sony mobile IP scheme problems can arise from the loss of *VipDelAmt* packet. The IETF scheme works well to handle packet losses. However as mentioned before, the lack of communication between the previous and current foreign agent may result in some packet loss. There have been some suggestions and proposals that allow the mobile host to notify its previous foreign agent of its new location so that the old agent can forward any incoming packets or old-binding caches to the new agent to address this problem. Such a scheme supports a smoother and more efficient handoff.

In our scheme, all management messages require a corresponding acknowledge. This ensures high reliability but comes at a cost of high overheads. Handoff procedures are smooth and the packets destined for an old BR still get routed to the current location of a mobile host via the RP if the host is still in the backbone. Or less a search is initiated to determine the AS holding the mobile host. In the latter case some latency is introduced but this is still tolerable when compared to the latency associated with triangle routing as in IETF scheme.

6. **Component Loss**: The Sony mobile IP scheme concentrates a substantial portion of mobile IP functionality in a home gateway, which makes it vulnerable to failure. The Columbia scheme on the other hand will generally recover from the losses of MSR after a cache time out. Assuming that the MHs that were connected to the down MSR can attach to an alternate MSR most users will have almost no indication that there was any problem. In the IBM mobile IP scheme the loss of a BAS can be a serious problem. If a BAS fails, any MHs registered with it will presumably register with a new BAS. However, hosts that were previously communicating with a MH attached to the down BAS will continue to unsuccessfully route packets through the down BAS. This situation will only resolve itself if the MH happens to send a packet to the other host causing its LSSR for the MH to be updated.

IETF mobile IP has a single point of failure in the home agent. If there is a home agent failure, all mobile hosts under the jurisdiction of this agent will remain inaccessible. The problem is amplified considering that a home agent can potentially serve a large number of hosts. This case applies if a foreign agent breaks down as well. Route optimisation mechanism might mitigate this problem for some existing connections, as the previous foreign agent might still be able to route the packet to the current foreign agent. However, a failure of an agent is a significant loss to network communications. There exists a recovery mechanism in this scheme in case of foreign agent breakage. In this situation, the mobile host would listen for signals sent from the foreign agent and after a timeout without hearing responses; it finds another agent to represent as its new care of address.

In our scheme, there are several key components. The effects of the failure of any of these components are different. Let us examine the failure on a case-by-case basis.
BBR Failure: Consider an example. A mobile host MH belongs to an autonomous system AS under a BBR BBR1. Assume that the MH is roaming and located elsewhere outside its AS. In this case, a failure of home BBR will not affect any nodes outside of the AS that wish to communicate with the target MH. Nodes within the AS however will not be able to communicate with any hosts outside as all communications must pass through the BBR and the BBR is the entity responsible for initiating as search process.

RP Failure: Consider a few scenarios. Scenario one is where the two communicating mobile hosts reside in different ASs. These mobile hosts can still communicate with each other as the search process is not affected by the failure of the RP and will yield correct location. In scenario two, the sending host in the AS and the target MH is the backbone. This has problems. Although the BR responsible for MH is operational the search process does not include the BR. Hence the MH remains inaccessible. In such situations the BBR must query the coordinator BR in the backbone. If this coordinator BR is aware of the location of mobile host it responds with the location details. Or else, this BR queries the All_CBR group to retrieve the location details. While the communication does not halt, this scheme introduces significant latency that may not be accessible for some applications. A third scenario is when both the communicating hosts are in backbone. The search process here is also adversely affected as it makes exclusive use of RP. To overcome this problem, the scheme requires the support of coordinator BRs. It has the usual problems that were cited above.

BR Failure: A failure of the BR will only affect all nodes that are currently residing in its cell.

7. Network Breakage: The Sony scheme requires a small amount of additional protection against breakage. For instance, if a mobile host MH is separated from its gateway, hosts in the same segment as the MH will be able to send packets to the MH as long as they happen to pass through a Sony router that has the MH’s current location cached. According to [Myles+ 93] this mechanism is not guaranteed to be successful as it depends on traffic patterns, cache time outs and the number of Sony routers in the topology. In Columbia and IETF schemes the continued operation for a MH in wide area environment after network breakage depends upon whether the home agent/MSR can be contacted or not. Component duplication is necessary in all these schemes to provide fault tolerance against network breakage.

In our scheme, in the backbone, a travelling host might experience some disruption if there is a network breakage and the MH is unable to establish a connection with a BR. However during the initial registration time if there is a network breakage and the MH is unable to access the RP, it may result in complete lack of communication for the MH. The solution would be for the MH to continue to hear beacon messages from other BRs that offer connectivity to the network. A break in the communication with the home AS does not prevent any other host outside this AS to have access to the mobile host. In case of network breakage when the RP becomes inaccessible, the network can utilise the fault tolerance support in the form of Coordinator BRs to provide the necessary connectivity.
8. **Processing Bandwidth:** For the IBM scheme processing bandwidth is quite high, as the options field for every packet must be processed. Existing routers generally have an optimised processing path for optionless packets and a much slower path with options [Myles+ 93]. Also every router must process every packet introducing substantial overheads. In fact in [Teraoka 92b] it was shown that the resultant overhead is at least 29% more than the equivalent IP processing. In the mobile IP scheme the triangle routing scheme introduces extra processing bandwidth. Without any route optimisation, packets must traverse through intermediate routers in the indirect path, where they will be individually queued, and processed before they are sent. This extra queuing and processing adds unnecessary overhead for packet transmission. Route optimisation scheme eliminates this problem, as it does not involve the routers in the indirect path.

While our scheme does not introduce any processing bandwidth like the IETF mobile IP scheme it does introduce some amount of protocol processing bandwidth as in Columbia Mobile IP scheme. This overhead occurs in the form of multicast messages in the form of a search process. Hence there is a trade off between processing bandwidth at the routing to protocol processing bandwidth. Since the search results can be cached, and the packets take optimum routing paths without any extra functionality in routers our scheme scales well when compared to other approaches.

9. **Impact on Existing Routing Protocols:** The Sony, IETF and IBM schemes introduce low overheads in terms of advertisement messages as they only advertise connectivity at the home gateway and MR respectively. In our scheme as well as in Columbia IP scheme the advertisements also occur over mobile subnets. However, our scheme just as the IETF scheme can be deployed with virtually no changes to the existing network protocols but unlike the IETF scheme our scheme offers optimum routes.

10. **Infrastructure:** In terms of infrastructure, the Sony scheme has some severe limitations. Firstly, it always requires facilities to acquire temporary IP address. It also makes it mandatory for a home gateway to be associated for each network (as in the IETF scheme) and finally it requires the participation of several routers to achieve optimum routing. In case of the Columbia scheme, it requires a MSR for every virtual mobile subnet. It also requires facilities for acquiring temporary address to facilitate communication with a remote roaming host (as in IETF scheme). The functionality of the BAS in the IBM scheme and the BR in our scheme are quite similar. Both the schemes require a BAS/BR to be located anywhere a mobile host is connected as well as a MR for every mobile subnet. Although it appears to be less practical to have a BAS located everywhere, however, with the proliferation of cellular infrastructure it can be concluded that this is not impossible to achieve. As the capability of normal base stations are enhanced to accommodate data traffic, no additional infrastructure will be necessary to support our scheme. Also the role of a home BSR is less intensive in our scheme when compared to other approaches, as it is not necessarily involved in the location management or routing functions for a mobile host.

11. **Migration Detection:** In the IBM and Columbia schemes, beacon messages are used to detect migration. Whereas the IETF scheme uses agent advertisement
messages. Our scheme uses beacon messages as in Columbia. These messages contain the addresses of the BR, BBR and RP to help achieve registration. It also serves the dual purpose of carrying ARP binding messages and certificates for secure registration.

12. **Address Usage**: The IBM schemes scales well for conserving IP addresses. Every mobile host has only one IP address ever assigned to it and only additional addresses are required for BAS in the topology. The Sony scheme scales poorly as each subnet must reserve a block of addresses equal in size to the maximum number of mobile hosts expected to be connected to that subnet at any one time. For Columbia scheme, in wired area network mode, requires the use of extra addresses in the form of transient addresses. Our scheme uses the care of address option wherein the BR and BBR need to act as the tunnel endpoints to messages that are generated to and from the mobile host. This does not impose any additional strain on already depleting resource of IP addresses and at the same time transfers the protocol processing overheads away from the power limited mobile hosts.

13. **Security**: None of the schemes have inherent support for security. There are proposals that have suggested security related extensions to IETF Mobile IP [Perkins 98, Schafer 2001]. Our scheme on the other hand, specifies a complete security framework using a public key infrastructure. Both control as well as data messages have security integrated into them. Our security framework is explained in the next chapter.

4.7 Observations

In conclusion, if a host in an autonomous domain AS and wishes to communicate with another host, then first it contacts its local BBR in its AS. The BBR checks whether the target host is in its AS. If not, it contacts the RP in the backbone. At any given time the RP maintains a list of all mobile hosts in the backbone. If a host in the backbone wishes to communicate with another host, it first contacts its local BR. The BR checks whether the target host is in its cache. If not, it contacts the RP in the backbone to find the BBR or BR under which the target host is residing.

In principle, there can be multiple backbone segments. Each of these segments will be governed by a RP and may be connected to one or more autonomous systems. This type of set up is called an area. A set of such areas is called an Area. The configuration within a area can be hierarchical or peer-to-peer in nature. In the former case, a single RP acts as a parent for the area. RPs of other areas provide summary advertisements to the parent indicating hosts currently located in their areas. The parent RP caches this information. A search for a host across an area involves querying the parent RP. However, in this case a single parent RP may become a bottleneck. In order to avoid this problem an alternate hierarchical model can be created by forming a parent peer group. In this case, a set of RPs form a group called the parent RP group. RPs representing other peer areas are called child RPs. Each child RP nominates a single RP from the parent group as its parent. It then sends summary advertisements of its location information to this parent. At a higher level, within the parent peer group, RPs exchange information with each other. In other words each RP within the parent group acts as a logical node representing a set of
peer areas. The peer areas represented by RPs within a parent group are known as child peer groups of that group. RPs of the parent group representing peer areas have the responsibility of formulating and exchanging advertisements with their peer nodes within the parent peer group to inform those nodes of mobile hosts’ reachability in their respective peer areas. In case the architecture is peer to peer, then all the RPs within a area simply form a multicast group. A search for a node across areas will involve sending a multicast query to all other RPs.

A mobile host may get disconnected while on the move. The disconnection may be transient in nature such as during handoffs or it may go down for an indefinite period of time. MH may also go into a "doze" mode. In such a case, MH can actually inform its peers before disconnecting, so that preventive action can be taken both at the MH and on the wired network to lessen the effect of disconnection on applications. To handle such cases, the BR maintains a timeout element $MHTimeout$. This information element indicates the amount of time elapsed since MH last contacted the BR. If the breakdown of MH is unprecedented and it does not get an opportunity to inform its BR of this event, then BR simply waits for $MHTimeout$ interval after which it simply purges this MH's entry from its database. RP is also notified after $MHTimeout$ so that the entry is deleted from RP's database. Before the expiry of the timeout, BR caches all packets destined to MH. MH also caches $MHTimeout$. When in doze mode, MH is aware of the buffer time it has in hand before getting back to BR. If the MH is in an autonomous domain, then it is the BBR that takes on the role of BR. A similar procedure to the above occurs and after timeout, BBR deletes the MH’s entry from its database and informs the RP.

4.8 Summary

In this chapter we proposed a new model for supporting mobility in networks. Unlike some other previous models proposed, this scheme offers some unique features by means of which optimum routing can be established between two communicating mobile hosts. For instance, in our scheme, even though we may have a dedicated home network associated with a roaming host, this network does not play a critical role in formulating routing decisions for the mobile host. The location information of a moving host is distributed and hence there is no single point of failure. We have also compared our scheme to four important mobile IP proposals and cited some advantages and disadvantages. While there is no single ideal protocol that can satisfy all the requirements of host mobility, our model comes close to fulfilling most of the necessary requirements.
Chapter 5

Security Architecture for DLM Model

5.1 Introduction

The security specifications for any mobile IP scheme can be considered from two perspectives: (1) the expectation of the mobile nodes to retain their network services and protect their communication when they visit the foreign subnets and (2) the expectation of the foreign subnets to protect their network resources and local traffic while they are visited by mobile nodes. [Zhao+ 97]. In other words, a mobile node (MH) must enjoy safe and persistent IP connectivity as much as this is permitted by the policies of its home and visited subnets. Persistency of IP connectivity means that the connections should be handed off correctly and quickly so that the MH can maintain its TCP sessions when it changes its network attachment point. Safety means traffic to and from the MN should enjoy a similar level of security (with respect to passive and active attacks) as it is on its home subnet.

Securing the DLM model presents some challenges. Unlike telephony networks, where a separate channel is reserved for control plane messages, mobile IP based networks such as the DLM network, utilize the normal public network used by the data messages to carry control traffic. For complete security, it is necessary to protect data in both the control plane as well as in the data plane in order to thwart security attacks. We propose Public Key Infrastructure (PKI) support for the DLM Architecture. While there may be provisions for facilitating security in the link layer level, we choose to ignore them as we are primarily concerned with securing end –to-end IP based traffic.

The principle objectives of the proposed security architecture for DLM are as follows.

**Privacy:** It must be able to support privacy of data and control packets using appropriate encryption techniques. This requires embedding security into the tunnelling mechanisms of IP. One important aspect of this service is to provide location privacy. Location privacy is important in cases where the knowledge of the location is as useful as the contents of the transmitted messages themselves. An issue that is tightly coupled with the provision of location privacy is that of anonymity. A mobile host must be able to roam around in a seamless fashion and at the same time must be able to conceal its real identity. At the same time various administrative entities in the network must be given an assurance that they are indeed providing services to a legitimate host and not a malicious one.

**Authentication:** It must be able to provide an authentication service between the various participating entities in the DLM model. In particular, authentication of control messages must be made mandatory. The following cases must be dealt with:

- Authentication between a mobile host and the RP
- Authentication between the mobile host and the BBR (Home as well as visiting BBR).
• Authentication between the BBRs (The BBRs representing two communicating mobile hosts).
• Authentication between the BBR and the RP.
• Authentication between the BR and the RP.

**Access Control:** It must be able to provide access control services. An example where access control can be used is in situations where the RP and the BBR must make a decision of whether or not to allow a mobile host from accessing resources and services at their respective ends. One principle requirement for the provision of access control is the verification of the identity of a mobile host. This process in principle may involve the home BBR of the mobile host.

**Integrity:** It must be able to provide integrity of both control and data packets. This is critical in situations where an attacker can gain control of location update messages in transit and modify its contents causing packets for a target mobile host to be redirected to different location.

While providing these services it is important to minimise the number of security related messages to reduce the overheads. For this reason, we embed security into existing control messages of the DLM model.

This chapter is organised as follows. Section 5.2 introduces the concept of key management and describes the requirements for a key management framework for DLM model. In section 5.3, we discuss the public key infrastructure for the DLM model. We also outline some important assumptions that we make in our design of security services for DLM model. In section 5.4 we propose secure end-to-end communication protocols in the DLM model. In section 5.5, we explain the significance of having certificates in the DLM environment and also explain the significance of X.509 certificate extensions with respect to the DLM environment. In Section 5.6, we provide our concluding remarks.

**5.2 Key Management for DLM architecture**

**5.2.1 Introduction to Key Management**

Key management deals with the secure generation, distribution, and storage of keys. Secure methods of key management are extremely important. Once a key is randomly generated, it must remain secret to avoid unfortunate mishaps (such as impersonation). In practice, most attacks on public-key systems will probably be aimed at the key management level, rather than at the cryptographic algorithm itself.

Users must be able to securely obtain a key pair suited to their efficiency and security needs. There must be a way to look up other people's public keys and to publicize one's own public key. Users must be able to legitimately obtain others' public keys; otherwise, an intruder can either change public keys listed in a directory, or impersonate another user. Certificates are used for this purpose. Certificates must be unforgeable. The issuance of certificates must proceed in a secure way, impervious to attack. In particular, the issuer must authenticate the identity and the public key of an individual before issuing a certificate to that individual.
If someone's private key is lost or compromised, others must be made aware of this, so they will no longer encrypt messages under the invalid public key nor accept messages signed with the invalid private key. Users must be able to store their private keys securely, so no intruder can obtain them, yet the keys must be readily accessible for legitimate use. Keys need to be valid only until a specified expiration date but the expiration date must be chosen properly and publicized in an authenticated channel.

5.2.2 Key Management Framework Requirements for DLM Model

In order to provide the above-mentioned security services, any proposed key management framework must satisfy the following requirements:

**Scalable Infrastructure:** The underlying key management infrastructure must be scalable to generate and dispatch long-term key parameters among any pair of network hosts. This primarily concerns key establishment. Without such an infrastructure it is not possible to secure authentication and communications channels between the participating hosts. In key establishment, public key cryptography is commonly used for key transport. An example of key transport is the use of the RSA algorithm [Rivest+ 78] to encrypt a randomly generated session key with the other party’s public key. The encrypted key is then sent to the recipient, who decrypts it using his private key. In order to bind these public keys to a host/user requires the presence of certificates and in order to establish a scalable system for generation and distribution of such certificates requires a Public key Infrastructure Support. We discuss the PKI framework for the DLM model in some detail in the next section.

**Light-Weight Key Generation Algorithm:** In order to generate keys, one must make use of key generation algorithms. These key generation algorithms must be lightweight i.e., they should not be process and resource intensive in nature as the mobile systems introduces a constraint on the availability of these two entities. An example of key generation using public key cryptography is the Diffie-Hellman Algorithm (DH) [Schneier+ 95]. This algorithm generates a session key based on public and secret information held by the communicating parties. This algorithm starts with the two parties exchanging the public information. Each party then mathematically combines the other’s public information along with their own secret information to compute a shared secret value. This secret value can be used as a session key or as a key for encrypting a randomly generated session key.

**Key Exchange Authentication:** The key exchange authentication can be done during the protocol or after protocol completion. Authentication of the key exchange during the protocol is provided when each party provides proof it has the secret session key before the end of the protocol. Encrypting known data in the secret session key during protocol exchange can provide proof. Authentication after the protocol must occur in following communications. In the DLM environment, authentication during the protocol is preferred so subsequent communications are not initiated if the secret session key is not established with the desired party thereby saving processing overheads and bandwidth.

**Key Exchange Symmetry:** In the DLM model, any host must be capable of initiating a key exchange message. These messages can cross in transit without affecting the key that is generated. This achieves Key Exchange Symmetry.
**Perfect Forward Secrecy**: This feature is provided by the key exchange protocol if disclosure of long-term cryptographic keying material (e.g. public signature keys) does not compromise previously generated keys. Past session keys will not be obtainable if the long-term key is compromised in perfect forward secrecy. This property must be reflected in the design of secure communication protocols between communicating hosts.

**Back Traffic Protection**: This feature is provided by the independent generation of each key in such a way that subsequent keys are not dependent on any previous key. Past session keys will not be obtainable if the current session key is compromised in back traffic protection. We take this factor into consideration in the design of secure end-to-end communication protocols between mobile hosts.

### 5.3 Public Key Infrastructure for the DLM Model

Our scheme uses public key technology to meet the security requirements of the DLM model. In particular, we suggest the use of a Public Key Infrastructure (PKI) framework for managing X.509 public key certificates and v.2 revocation lists [Housley+ 97] issued to hosts. The main reason to use PKI technology is for scalability as in [Zao+ 99]. In order to support global Internet mobility, a technology must be able to establish shared secrets among a large set of hosts spread across multiple Internet domains. Also the hierarchical trust relations among certification authorities can be exploited by mapping them onto the domain-based topology of DLM model especially in situations where our model is scaled further using hierarchical RP scheme as described in section 5 in the previous chapter.

First, it is essential to understand the PKI entities and their relation to the DLM environment. The three principal entities are:

- **Mobile Hosts/ Static Hosts**: This set includes the mobile roaming hosts as well as stationary hosts and administrative entities such as the BR, the BBR, and the RP who have the capability of generating public/private key pairs and get certified using a trusted Certification Authority.

- **Certification Authority (CA)**: An entity that verifies the identity of a entity, allocates a PKI distinguished name to the entity, and verifies the correctness of information concerning that entity by signing a public key certificate for that entity. Public keys are stored and distributed in the form of certificates. The signature of the CA ensures that the specified public key belongs to someone or something with a specified name. Any node that has the public key of the CA can verify the signature of the certificate and thereby validate and obtain the public key in the certificate. A certificate typically contains a serial number, version number, signature algorithm identifier, the name of the issuing CA, the validity period of the certificate, the name of the node, the public key of the node as well as the CA’s signature on all of the above fields. In our architecture, the CA functionality is co-located with the BBR/RP.
**Directories**: The directories act as repositories for certificates. In our scheme the directory service is co-located with the BBR/RP.

Logically, the DLM model is divided into different domains called autonomous systems connected by a backbone network. Moreover, a single administrative authority called the autonomous system controller, which typically resides in the BBR, governs each autonomous system. Similarly, the backbone falls under the jurisdiction of a single administrative authority called the RP. So we can divide this architecture into different certification domains where the certification domain either maps on to an autonomous system or to a backbone. In the backbone, the certification authority can be co-located with the RP. Each autonomous system contains a CA. This CA can either be co-located with the BBR or can exist as a separate logical entity.

In the case where two nodes wishing to communicate with each other reside within the same certification domain, they can validate each other’s public key certificate using their CA’s public key. When two nodes reside in two different domains, there are two ways for achieving the inter-domain certification.

**Hierarchical structure**: The first approach uses the hierarchical certification model. For instance, we can have a root CA similar to the IETF’s Internet Policy Registration Authority. This root CA certifies the public key of every CA that is immediately beneath it and this process is followed recursively. At the bottom most level, the leaf CA certifies the public keys of nodes below it. In order for the nodes to communicate, the source node sends to the destination node a full certification path, that is, its own certificate along with every superior CA’s certificate up to the root CA. The assumption here is that every node knows the root CA’s public key.

**Cross Certification**: Alternatively, if there exists a cross-certification mechanism between the CAs, then this helps to shorten the certification path. Cross-certification involves two CAs certifying each other directly without going through their higher-level CAs. Therefore, the greater the number of such cross certification links, the greater the likelihood that the certification paths will be shorter. However cross-certification between two CAs involves prior communication between them, which is a part of the inter-CA registration process. The first approach involves registration of all CAs with the root CA. An organisation such as the ITU could be the guardian of such a root CA. The second approach enables organisations to directly negotiate and cross-certify other organisations’ CAs. This is particularly useful when nodes in the two organisations will have to communicate with each as part of their normal business needs. Furthermore, within a large organisation (or a multinational organisation), it is likely that there will be an internal hierarchy of CAs with an organisational root CA.

Several efforts are on the way to establish a hierarchical CA structure. While this is happening, the natural step is to establish some leaf CAs (be they large organisations or certain specific organisations that provide certification services) and develop cross-certification links. This provides an incremental way of introducing the public key infrastructure into the DLM networks.
Hence we propose a combination of the hierarchical and cross certification approaches. This results in a hybrid structure, which forms the building block of the DLM PKI. The following principles and assumptions are used in the formulation of our security model:

- Every host is initially registered in some AS. The host upon registration receives a certificate that is signed by some CA that is local to that AS. This process is discussed later.
- BBR acts as a CA for its AS. RP acts as a CA for the backbone. Directory services are co-located with BBR and RP.
- Each BR, BBR and RP has a public key. RP maintains a cache of the public keys of BBRs and BRs. BBRs maintain a cache of public keys of other BBRs, the RP as well as the neighboring BRs.
- Each BR/BBR and RP is trusted for authenticating a mobile user based on public key information and for maintaining the relevant security-related information.

5.4 Secure End-to-End Protocols and Message Flow Semantics

In this section, we describe the protocols necessary to facilitate secure communication between the nodes and the PKI authorities. In the protocol definitions of the following sections we use the following notation:

- **MH**: Identity of MH.
- **CA**: Identity of CA.
- **BBR<sub>x</sub>**: Identity of the current mobile router for MH.
- **T**: Timestamp and nonce to indicate the timeliness of message.
- **N**: Nonce indicating the freshness of message.
- **P<sub>K-X</sub>**: Public key of principal X, available in a certificate.
- **Signing<sub>Algo</sub>**: The algorithm used by the MH to sign the signed portion of the message.
- **Cert<sub>X</sub>**: Certificate of principal X generated by some CA.
- **Cert<sub>Path</sub>**: This contains the certification path from the root to a principal.
- **CRL<sub>Path</sub>**: Certification revocation List (CRL) path from a principal to the root. This also includes the current copy of the CRL issued by the parent CA.
- **{…}<sub>S<sub>K-X</sub></sub>**: This notation signifies that the contents within {…} are hashed and signed using the private key of principal X, S<sub>K-X</sub>.
- **P<sub>K-X</sub>[…]**. This notation signifies that the contents within […] are encrypted under the private key of the principal X, P<sub>K-X</sub>.
- **loc<sub>X</sub>**: Current location of an entity X.

5.4.1 Initial Registration/Certification

This case refers to a situation where the mobile host has not had any previous contact with the PKI. A typical scenario involves a person joining the organization. In our model, an AS may map on to an organization network. If a masquerade attack is to be prevented this scheme will require the use of some out-of-band mechanism to establish an initial session key between the host and its CA. In this case, the BBR acts as a CA. This session key is then used to protect the registration and certification messages between the host and the CA. To prevent eavesdropping the host can use the
public key of the CA, recovered from the beacon message from its local router. This beacon message contains the public key of the CA. In this case, we assume that the mobile host has the capability to generate a public key-private key pair. Having done this, the host sends a certification request \([\text{CERT}_\text{REQ}]\) along with its public key. The purpose of this message is to request the CA to generate a certificate for the concerned mobile host. The host signs the message using its private key and places its public key outside the signed message for CA to verify its signature. For preventing eavesdropping only, the portion of the message shown below as plain text is encrypted using the public key of the CA. If we are to prevent both masquerading and eavesdropping the plain text portion is encrypted using the previously established session key.

(1) \(\text{MH} \rightarrow \text{BBR}: \text{MH, CA, T, N, } P_{k_{\text{MH}}}, \text{Cert_Details, Signing_Algo., } \{\text{MH, BBR, T, N, } P_{k_{\text{MH}}}, \text{Cert_Details}\}S_{K_{\text{MH}}}

Note that in this case, the signed portion includes the nonce, timestamp, identities of the MH and CA, and Cert_Details. The nonce and timestamp are used to thwart replay attacks whereas the identities are necessary for authentication purposes. The parameter Cert_Details contains information pertaining to the mobile user, which is used by the CA to generate the certificate for the user.

The CA, upon receiving this message can use the public key of the MH to validate the signed element. If the check is successful, then the CA responds by generating the certificate and sending it to the MH.

(2) \(\text{BBR} \rightarrow \text{MH}: \text{BBR, MH, } T_1, \text{ N, } N, P_{k_{\text{RootCA}}}, \text{Cert_MH, Cert_Path, CRL_Path, } \{\text{BBR, MH, } T_1, \text{ N, } N\}S_{K_{\text{BBR}}}

The nonce N is re-used by the CA to bind this response to the request message (1). The signed portion of this message it includes the nonce, timestamp, and identities of the MH and CA. Even though we consider a two level hierarchy, there may be still be situations where you have a hierarchy of CAs existing with an AS.

5.4.2 Authentication and Secure End-to-End Communication

As mentioned before, in a mobile IP environment, the messages transmitted over the wireless links are not only susceptible to eavesdropping but it is also possible for any user to access the mobile network using a false identity. Therefore in order to secure such networks, we must use encryption techniques. The encryption technique serves the dual purpose of providing confidentiality of the messages sent over unreliable wireless links as well as for the authentication between two communicating parties. In our system, we use a combination of public key and symmetric key based approaches to achieve the same. In this system, a common key called the session key is shared between the entities before any communication session begins and later these session keys are used to encrypt the data.

End-to-end IP security refers to the security of the communication between the mobile host and the correspondent host. An obvious approach to achieving end-to-end IP security is to utilise the existing IPSec framework. But the disadvantage of this
approach is that it puts too much burden on the mobile host [He+ 00]. We use a hybrid approach involving both symmetric key and public key based systems. We use symmetric key based cryptosystems for securing communications on an end-to-end basis as it is computationally less intensive than public key systems. Consider the scenarios described in section 2.3 in the previous chapter. We will integrate security into each of these scenarios based on this hybrid scheme. From the communicating hosts’ perspective securing the end-to-end path between the two hosts is of primary concern. Therefore, the main objective of the protocols described below is to provide mutual authentication between mobile hosts and to establish a secret shared conversation key between them in a secure fashion.

**Case 1: Mobile host and the correspondent host are in the same AS**

**Protocol Objective:** Secure communication between a correspondent host (CH) and the mobile host (MH) when both reside in the same autonomous system.

Assume that MH is under mobile router BBR\(_x\) in the autonomous system AS\(_1\) and the CH is under the mobile router BBR\(_y\) in autonomous system AS\(_2\). MH initiates a conversation with CH. There are three options available, which differ in the generation of the session key.

**Case 1-Option One: MH and CH contribute to Session Key**

1. **MH→BBR\(_x\):** Cert\(_{MH}\), \(T_1\), \(N_1\), \(P_{K-BBR_x}[CH]\), \{MH, CH, \(T_1\), \(N_1\), Flag\}\(S_{K-MH}\)

The purpose of protocol step (1) is for MH to authenticate itself to its BBR and pass on a request to communicate with a correspondent host CH in a secure fashion. In order to achieve these objectives, MH constructs a message that includes its identity, certificate, an encrypted element, and a signed element to the mobile router BBR\(_x\). The encrypted element contains the identity of the correspondent host encrypted under the public key of BBR\(_x\). The purpose of this element is to securely convey the identity of the target host CH with whom the communication session is being set up. The presence of this element achieves secrecy of the identity of the intended correspondent host and since it is encrypted under the public key of BBR\(_x\), it also proves that the message is for that BBR and no one else. The purpose of the signed element is thwart repudiation attacks by providing the service of data origin authentication. The use of a timestamp and nonce is to defend against replay attacks. MH might indicate beforehand whether it has the certificate of the correspondent host through some previous communication session using the flag field. This element is hashed and signed using the private key of the MH.

Upon receiving this message, BBR\(_x\) must do the following. It must validate this message. It must then furnish the current location details of the concerned correspondent host to the MH with appropriate security parameters necessary to establish for MH to initiate a secure session with the CH. When BBR\(_x\) gets the message of step (1), BBR\(_x\), verifies the certificate of the MH and then uses the public key recovered from the certificate to verify the signed element to authenticate the sender MH and to also check the integrity of the message. It checks the timestamp and nonce to establish the freshness of the message. It decrypts the encrypted element
using its private key. This proves that BBR\textsubscript{x} was in fact the intended recipient of this message.

(2) \textbf{BBR}\textsubscript{x} → MH: BBR\textsubscript{x}, MH, T\textsubscript{2}, N\textsubscript{1+1}, P\textsubscript{k-MH}[\text{Cert\_CH, locCH, CH}, \{MH, CH, T\textsubscript{2}, N\textsubscript{1}, N\textsubscript{1+1}\}]\textsubscript{S\textsubscript{K-BBR}}

In response to step (1), in step (2) the BBR\textsubscript{x} sends an encrypted element and a signed element to MH. The purpose of the encrypted element is to securely send the certificate of CH along with its location details to MH. The reason for encryption is to provide location privacy. The signature element provides the certificate of the correspondent host as well as its location in a secure fashion to the MH. Location privacy is achieved as the location details are encrypted under the public key of the MH. The motivation to include the location details is for the following reason. While the location details can be retrieved using regular broadcasting and multicasting schemes it generates unnecessary overheads. Besides BBR\textsubscript{x} is the only entity that keeps track of the current location of mobile hosts at all times. The signature element is created to thwart repudiation attacks. It also binds the request of the MH in step to this response in step (2) by including MH’s nonce from step 1. Nonces and a timestamp are included to prevent replay attacks. These elements contain the certificate of the correspondent node (CH) along with location information details of the CH.

If MH must frequently communicate with CH, then it may cache this certificate. Later when it generates a request to locate the MH it might indicate using the flag field that it already has the certificate of the CH stored locally.

MH now generates the following message to CH.

(3) MH → CH: MH, CH, Cert\_MH, P\textsubscript{K-CH}[K\textsubscript{S1}, T\textsubscript{3}, N\textsubscript{2}], \{MH, CH, T\textsubscript{3}, N\textsubscript{2}, Algo\_Details\}]\textsubscript{S\textsubscript{K-MH}}

The main objective of message flow in step (3) is for MH to initiate a secure end-to-end communication session with CH. It also needs to generate its share of the session key and convey this share in a secure fashion to the CH. In the process, it must also authenticate itself to the CH. In step (3), MH sends its identity, certificate, an encrypted element, and a signed element to the CH. The purpose of the encrypted element is to convey MH’s share of the session in a secure way by encrypting it under the public key of the CH. The purpose of the signed element is to thwart repudiation and replay attacks. It also conveys the algorithm details used for generation of the session key at its end. The signed element conveys to CH the node MH’s contribution to the session key (K\textsubscript{S1}). This element is signed using the private key of MH.

Upon receiving this message CH verifies the certificate and decrypts the encrypted element using its private key. It then uses the public key recovered from the certificate to verify the signed element to authenticate the sender MH and to also check the integrity of the message. It checks the timestamp and nonce to establish the freshness of the message. This proves that CH was in fact the intended recipient.
CH generates the following response to MH.

\[(4) \text{ CH} \rightarrow \text{MH: MH, CH, } P_{K\cdot\text{MH}}[K_{S2}, T_4, N1, N2], \{\text{MH, CH, } K_{S2}, T_4, N2\}S_{K\cdot\text{MH}}\]

The purpose of step (4) is for CH to authenticate itself to the MH and provide its share of session key details. The purpose of the encrypted and signed element is similar to that used in step (3). The nonce used in step (3) is repeated in step (4) to bind the two messages. Replay attacks are prevented through the use of nonces and timestamps.

**Case 1-Option Two: MH and CH contribute to Session Key with BBR as intermediary**

In the previous option, after the BBR conveys to MH the public key details of CH it plays no further role. The objective of this protocol is for the BBR to act as the trusted arbiter to authenticate both the parties prior to the communication. However the BBR is not trusted to know the session key details. Step (1) is the same as in option 1.

\[(2) \text{ BBR} \rightarrow \text{CH: BBR, CH, } T_2, N_2, P_{K\cdot\text{CH}}[\text{MH, Cert}_{\text{MH}}] \{\text{CH, MH, } T_2, N_2\}S_{K\cdot\text{BBR}}\]

The objective of message flow in step (2) is for the BBR to securely convey to CH MH’s intention of having a communication session with CH. This message flow includes an encrypted element as well as a signed element. The purpose of the encrypted element is to securely pass on the sender’s details to the recipient. It includes the identity of MH as well as its certificate and this element is bound to the signature element by including the identity of the MH in the signature element hashed and signed by the private key of the BBR. A timestamp and a nonce are included in the signature element to prevent against replay attacks.

In this message flow sequence the BBR is trusted to act on behalf of a mobile host. In some cases, where the level of trust is not high, the BBR must include the original request of the MH in step (2) for CH to believe that the MH is the intended sender of the message.

\[(3) \text{ CH} \rightarrow \text{BBR: CH, BBR, Cert}_{\text{CH}}, T_3, N_{2+1}, P_{K\cdot\text{BBR}}[\text{MH, } T_2, T_3], \{\text{MH, CH, T}_3, T_2, N_2, N_3\}S_{K\cdot\text{CH}}\]

The objective of protocol step (3) is for CH to supply its credentials in a secure manner to MH via the BBR. Upon validating the message flow (2), in message flow (3), BBR generates an encrypted element as well as a signed element. The encrypted element includes the identity of MH encrypted under the public key of BBR. The purpose of this encrypted element is to achieve the following goals. CH can securely convey that it received the request of BBR at time $T_2$, and it indeed wants to have a communication session with MH. The signed element includes the timestamps from message flows (2) and (3) to bind them. This signed element is hashed and signed using the private key of the CH.
(4) **BBR → MH:** BBR, MH, T₄, N₁, N₄, P_{K_{MH}}[locCH, Cert_CH] {locCH, CH, MH, T₄, N₁, N₄ }S_{K_{BBR}}

The objective of message flow (4) is for the BBR to securely convey the credentials of CH along with its location details to MH. For this purpose, it includes an encrypted element that contains the CH location information as well as its public certificate. While the public key certificate of CH can be left out in the open since its identity is included in the signature element, however, revealing this would reveal the identity of the other communication end point. The signed element includes the timestamps from message flows (1) and (4) to bind them. This signed element is hashed and signed using the private key of the BBR.

(5) **MH → CH:** MH, CH, T₅, N₅, P_{K_{CH}}[K_{S₁}], {K_{S₁}, MH, CH, T₅, N₅}S_{K_{MH}}

The objective of message flow (5) is for the MH to generate a direct communication session with the CH. Upon validating the origin and content of message flow (4), MH generates a message that includes a signed encrypted element and a signed element. The purpose of the encrypted element is to securely convey MH’s contribution to the session key by encrypting it under the public key of the CH. The signed element also includes the session key of MH.

(6) **CH → MH:** CH, MH, T₆, N₅, N₆, P_{K_{MH}}[K_{S₂}]
{K_{S₂}, CH, MH, T₆, N₅, N₆}S_{K_{CH}}

The purpose of message flow (6) is for the CH to convey its contribution off the session key to MH in a secure fashion. The signed and encrypted elements have goals similar to that of Step (5).

**Case 1-Option 3 MH and CH contribute to Session Key with BBR as trusted intermediary**

In this option, the BBR is trusted to know the session key contributions of both CH and MH. Messages 1 and 2 are the same as in option 2

(3) **CH → BBR:** CH, BBR, Cert_CH, T₃, N₂, P_{K_{BBR}}[K_{S₁}, MH], {K_{S₁}, MH, CH, T₃, N₂, N₃}S_{K_{CH}}

The objective of protocol step (3) is for CH to supply its credentials in a secure manner to MH via the BBR. However, unlike the previous option, the BBR is trusted to pass on CH’s contribution to the session key. The message flow includes a signed and an encrypted element. The encrypted element includes the identity of MH encrypted under the public key of BBR. The purpose of this encrypted element is to achieve the following goals. CH can securely convey its contribution of session key and that it indeed wants to have a communication session with MH. The signed element includes the timestamps from message flows (2) and (3) to bind them. This signed element is hashed and signed using the private key of the CH.

(4) **BBR → MH:** BBR, MH, T₄, N₁₊₁, P_{K_{MH}}[K_{S₁}, locCH, Cert_CH] {K_{S₁}, locCH, CH, MH, T₄, N₁, N₄}S_{K_{BBR}}
The objective of protocol step (4) is for the BBR to securely convey the credentials of CH along with its location details to MH. For this purpose, it includes an encrypted element that contains the CH location information as well as its public certificate. The signed element includes the timestamps from message flows (1) and (4) to bind them. This signed element is hashed and signed using the private key of the BBR. Note that prior to this message flow, MH was unaware of the location details of CH.

\[ (5) \quad \text{MH} \rightarrow \text{BBR}: \text{MH, BBR, T}_5, N_{1+2}, P_{K-BBR} [K_{S2}, \text{CH}], \{K_{S2}, \text{MH, CH, T}_5, N_1, N_4, N_5\} S_{K-CH} \]

The objective of protocol step (5) is for CH to supply its credentials in a secure manner to MH via the BBR. The message flow includes a signed and an encrypted element. The encrypted element includes the identity of CH encrypted under the public key of BBR. The purpose of this encrypted element is to securely convey MH’s contribution to the session key to CH. The signed element includes the timestamps from message flows (1), (4) and (5) to bind them. This signed element is hashed and signed using the private key of the MH.

\[ (6) \quad \text{BBR} \rightarrow \text{CH}: \text{BBR, MH, T}_4, N_{2+2}, P_{K-MH} [K_{S2}, \text{locMH}], \{K_{S2}, \text{CH, locMH, MH, T}_4, N_{2+2}\} S_{K-BBR} \]

The objective of protocol step (6) is for the BBR to securely convey the credentials of CH along with its location details to MH. This message flow includes an encrypted element that contains the MH location information as well as its public certificate. The signed element includes the timestamps from message flows (3) and (6) to bind them. This signed element is hashed and signed using the private key of the BBR. Note that prior to this message flow, CH was also unaware of the location details of MH.

**Case 2: Mobile host and the correspondent host are in different AS**

**Protocol Objective**: Secure communication between a correspondent host (CH) and the mobile host (MH) when both reside in different autonomous systems.

\[ (1) \quad \text{MH} \rightarrow \text{BBR}_x: \text{Cert}_\text{MH}, T_1, N_1, P_{K-BBR_x} [\text{CH}], \{\text{MH, CH, T}_1, N_1\} S_{K-MH} \]

The purpose of protocol step (1) is for MH to authenticate itself to its BBR and pass on a request to communicate with a correspondent host CH in a secure fashion. The details are similar to Case One-Option one.

\[ (2) \quad \text{BBR}_x \rightarrow \text{ALL}_2\_\text{BBR}: \text{BBR}_x, \text{ALLBBR}, \text{Cert}_\text{BBR}_x, T_2, N_2, P_{K-\text{ALLBBR}} [\text{CH}], \{\text{CH, T}_2, N_2\} S_{K-\text{BBR}_x} \]

Since the CH is not in this AS, BBR\_x must initiate a search. The objective of the message flow in step (2) is to send a search message to the ALL\_BBR multicast group in a secure fashion. The message flow contains an encrypted element as well as a signed element. The encrypted element contains the identity of target CH being searched and this encrypted using ALL\_BBR multicast group key P_{K-ALL\_BBR} The
purpose of the signed element is to thwart repudiation and replay attacks. For this reason, it contains the identity of CH along with appropriate nonce and timestamp hashed and signed under the private key of BBR.

\[(3) \quad \text{BBR}_Y \rightarrow \text{BBR}_X: \text{BBR}_Y, \text{BBR}_X, \text{Cert}_{\text{BBR}_Y}, T_3, N_{2+i}, P_{K-\text{BBR}_X} [\text{CH, Cert}_C] \{\text{CH, BBR}_Y, T_3, N_2, N_3\} S_{K-\text{BBR}_Y} \]

Upon receiving the multicast message, each BBR performs a search in its database to locate CH. In this case, BBR \(Y\) tracks down the entry for the concerned CH. The objective of the message flow in step (3) is for BBR \(Y\) to securely send a response message claiming to be holding CH. It includes, as an encrypted element the identity of CH and its certificate. The signed element includes the identities of CH and BBR \(Y\) along with nonces from message (2) and (3) to prevent replay attacks and to bind the request with the response.

The protocol then proceeds from message 2 of case1, option 1. A brief explanation is as follows. In step (4), BBR \(x\) securely hands the certificate of CH to MH by encrypting it under the public key of MH. MH verifies the signature of the signed element, decrypts the message and retrieves the public key of CH. In step (5), MH generates a message for CH. The encrypted element is encrypted under the public key of CH and the signed element is signed using the private key of CH. Using this message MH authenticates itself to CH and conveys its portion of session key \((K_{S1})\) to CH. In step (6), CH authenticates to MH and supplies its portion of session key \((K_{S2})\).

The BBRs are used to exchange the public key certificates of the nodes. Once each node has the other node’s public key certificate, they can initiate a secure session between them by using a session key.

**Case 3: Mobile Host in the Backbone and the Correspondent Host in an AS**

**Protocol Objective:** Secure communication between a correspondent host (CH) and the mobile host (MH) when MH resides in the backbone and the CH resides in an autonomous system.

The steps here are the same as in case two. The only difference being that MH now sits under BR in backbone and CH resides in AS. The search technique involved is the same as shown in case two. The protocol is the same, except that BBR \(x\) is replaced with BR.

**Case 4: Mobile Host and the Correspondent Host in the Backbone**

**Protocol Objective:** To elucidate the protocol steps necessary for secure communication between a correspondent host (CH) and the mobile host (MH) when both reside in the backbone.

If MH and CH are under the same BR then this is the same as case 1, except that BBR is replaced with BR. If CH is not registered with the BR we proceed as follows (note that step 1 is the same)

\[(1) \quad \text{MH} \rightarrow \text{BR}_X: \text{Cert}_C, T_1, N_1, P_{K-\text{BBR}_X}[\text{CH}], \{\text{MH, CH, T}_1, N_1\} S_{K-\text{MH}}\]
The purpose of protocol step (1) is for MH to authenticate itself to its BR, BR_{X}, and pass on a request to communicate with a correspondent host CH in a secure fashion. In order to achieve these objectives, MH constructs a message that includes its identity, certificate, an encrypted element, and a signed element to the mobile router BR_{X}. The encrypted element contains the identity of the correspondent host encrypted under the public key of BR_{X}. The purpose of this element is to securely convey the identity of the target host CH with whom the communication session is being set up. The presence of this element achieves secrecy of the identity of the intended correspondent host and since it is encrypted under the public key of BR_{X}.

The purpose of the signed element is to thwart repudiation attacks by providing the service of data origin authentication. The use of timestamp and nonce is to defend against replay attacks. MH might indicate beforehand whether it has the certificate of the correspondent host through some previous communication session using the flag field. This element is hashed and signed using the private key of MH.

(2) \[ \text{BR}_{X} \rightarrow \text{RP}: \text{BR}_{X}, \text{RP}, \text{Cert}_{\text{BR}_{X}}, T_2, N_2, P_{K-\text{PR}}[\text{CH}] \{\text{CH}, T_2, N_2\}S_{K-\text{BR}_{X}} \]

Upon validating the message flow of Step (1), BR_{X} generates a LOCATE_MH message to the RP requesting it to identify the current location of CH. The purpose of the encrypted element in this message is to request the location and identity of the target host in a secure fashion. In other words, it achieves location privacy. The certificate of BR_{X} is included for the RP to validate the signed element. The signed element is hashed and signed using the private of BR_{X}.

(3) \[ \text{RP} \rightarrow \text{BR}_{X}: \text{RP}, \text{BR}_{X}, T_3, N_2+1, P_{K-\text{RP}}[\text{CH, locCH, Cert}_{\text{CH}}] \{\text{CH, locCH, T}_3, N_2, N_3\}S_{K-\text{BR}_{X}} \]

The purpose of message flow (3) is for the RP to respond with the location details of CH in response to BR_{X}’s request of message flow (2). The RP’s response includes an encrypted and a signed element. The encrypted element includes the location details of CH along with its certificate. This certificate will be invariably used by MH to set up a secure communication session with the CH. The exchange then proceeds from message 2 of case 1, option 1.

5.4.3 Observations

The protocol steps mentioned above provide confidentiality of message contents using encryption element encrypted under the public key of the recipient and appropriate session keys. In fact, the use of session keys restricts the viewing of the actual data traffic to only the communicating parties. Location privacy is achieved by placing the location details of the concerned target within the encrypted element. In the above protocol steps, we achieve location privacy by hiding the location details of the target hosts from attackers. However, the administrative entities besides the communicating parties are aware of the location details of the host. While it is possible to design protocols to achieve location privacy from the administrative entities, such a facility does not conform to actual rules of operation of the model. In some cases, it may be necessary to hide the locations details of the communicating parties from one another. This can be achieved by the administrative entities simply withholding the location information.
details and not including them in the response messages sent back to the communicating hosts. Message integrity protection and protection against replay attacks are achieved using the signed element. The element is hashed and signed using the private key of the sender preventing an attacker from modifying the contents of the purported message. The signed element in each message also includes appropriate nonces and timestamps to ensure the freshness of the message and to bind a request to a response and thereby prevent replay attacks. The presence of a signed element also thwarts repudiation attacks. Also note that in these protocols, we have carefully separated the information, which needs to be signed (for integrity and authentication) from that which needs to be encrypted (Confidentiality). It is particularly important to adhere to this design principle as mixing these two aspects leads to lack of clarity in the protocol design which is often an important source of protocol flaws [Varadharajan+ 95]. The storage of keys within mobile hosts is also an issue that must be addressed. Such keys can be stored in tamper-proof smart cards with appropriate interface to the mobile host. Requiring a key/password to activate the smart card can further strengthen the scheme.

Although we do not address the issue of anonymity in this protocol design it is worth discussing this service here. There are two degrees of anonymity that can be achieved in this model. In the first degree of anonymity, the mobile users remain anonymous with respect to one another when they engage in the actual communication session. This can be achieved by associating aliases to the mobile entities. However, the administrative entities are not only aware of this alias to true identity mapping but also authenticate this piece of information before undertaking procedures to establish the communication session between the concerned parties. It is evident that for such a scheme to work the nodes must explicitly trust the administrative authorities. In the second degree of anonymity, the actual identities of the communicating parties are even concealed from the administrative entities of the network. This is critical in situations where by simply tracking the movements of a mobile entity, security is significantly compromised. For such situations we suggest that the home BBR of the concerned user generate a token that acts as a pseudo certificate for each alias used by the mobile user. This token is duly signed by the home BBR of the mobile user. In order to achieve the second degree of anonymity an explicit trust relationship must be established between the home BBR of the mobile user and other administrative entities in the network. Another orthogonal issue is to address anonymity with respect to either host level anonymity or user identity level anonymity. Since our protocols operate at the host level, we allude to the former case here. We discuss the latter case in chapter 6. In this case, the administrative home domain might associate a series of IP addresses with a mobile host. Up to 4 IP addresses can be assigned for each network adaptor card. One such address could be public and others could be private to conserve IP address space. Each pseudo certificate would then be generated for each private address associated with the mobile host. This calls for binding multiple IP addresses to a single interface or multiple IP addresses can be bound to multiple interfaces. In such cases each roaming host must be configured as a server. Also issues related to tunnelling of datagrams and firewall traversal across multiple domains must be resolved.
5.5 Extensions to X.509 Certificates

As mentioned above, our scheme makes use of X.509 certificates. The primary motivation was its support for hierarchical trust relationships that can be easily embedded over the administrative domain based approach of the DLM model. The other important reason being the feature of flexibility offered by this certificate format. The X.509 certificate format defines extension fields that can be carefully exploited by accommodating useful information such as policy information to conduct access control, trust relationships between domains, and other useful key parameters. Also X.509 certificates are the current industry standard for digital certificates. The X.509 standard is supported by a number of protocols, including Privacy Enhanced Mail (PEM) [RFC 1421-24], Public Key Cryptography Standards (PKCS) [RFC 2985], Secure Hyper Text transport Protocol (S-HTTP) [Rescorla+ 96], and Secure Socket Layer (SSL) [Freier+ 96]. All X.509 certificates have the following data, in addition to the signature:

**Version**
This identifies which version of the X.509 standard applies to this certificate, which affects what information can be specified in it. Thus far, three versions are defined.

**Serial Number**
The entity that created the certificate is responsible for assigning it a serial number to distinguish it from other certificates it issues. This information is used in numerous ways, for example when a certificate is revoked its serial number is placed in a Certificate Revocation List (CRL).

**Signature Algorithm Identifier**
This identifies the algorithm used by the CA to sign the certificate.

**Issuer Name**
The X.500 name of the entity that signed the certificate. This is normally a CA. Using this certificate implies trusting the entity that signed this certificate. (Note that in some cases, such as root or top-level CA certificates, the issuer signs its own certificate.)

**Validity Period**
Each certificate is valid only for a limited amount of time. This period is described by a start date and time and an end date and time, and can be as short as a few seconds or almost as long as a century. The validity period chosen depends on a number of factors, such as the strength of the private key used to sign the certificate or the amount one is willing to pay for a certificate. This is the expected period that entities can rely on the public value, if the associated private key has not been compromised.

**Subject Name**
The name of the entity whose public key the certificate identifies. This name uses the X.500 standard, so it is intended to be unique across the Internet. This is the Distinguished Name (DN) of the entity, for example,

  CN=Java Duke, OU=Java Software Division, O=Sun Microsystems Inc, C=US

(These refer to the subject's Common Name, Organizational Unit, Organization, and Country.)
Subject Public Key Information

This is the public key of the entity being named, together with an algorithm identifier, which specifies which public key crypto system this key belongs to and any associated key parameters.

X.509 Version 1 has been available since 1988, is widely deployed, and is the most generic. We use X.509 version 3 certificates. In addition to the above-mentioned fields this type of certificate supports the notion of extensions, whereby anyone can define an extension and include it in the certificate. Some common extensions in use today are: KeyUsage (limits the use of the keys to particular purposes such as "signing-only") and AlternativeNames (allows other identities to also be associated with this public key, e.g. DNS names, Email addresses, IP addresses). Extensions can be marked critical to indicate that the extension should be checked and enforced/used.

There are some design choices available with the additional certificate fields that are worth a discussion. To be specific, the following additional fields have some significance from the DLM model perspective:

• **SubjectAltName** (Subject Alternate Name): An extension to Subject Name field.

• **IssuerAltName** (Issuer Alternate Name): An extension to the Issuer Name field.

• **Name Constraints**: An extension used only in CA certificates to specify the subject name space within which the CA is authorised to issue certificates.

• **Certificate Policies**: Specifies the subnet/domain affiliations.

• **Policy Map**: Is used with cross certification to establish bindings between certificate policies that can be considered equivalent along different certification paths.

• **Policy Constraints**: This extension is restricted to CA certificates and is used to enable/inhibit policy mapping and requires specific policy identifiers to exist in every certificate along the certification paths.

The exact usage of these fields with respect to the DLM model is still being investigated. The following discussion brings forward the issues that surfaced out of this investigation.

The subject name as mentioned above may be the X.509 distinguished name of the user. It can also be a canonical domain name of the host, which serves as an unambiguous pointer to a DNS entry in a DNS server. The other option available is to embed the IP address of the host/interface in the certificate. We prefer the latter option for the following reasons. All the protocol messages between the participating entities in the DLM scheme operate at the IP level. They use IP addresses to identify the DLM entities. Some other advantages of associating certificates with IP addresses can be identified [Zao+ 99]. The certificates can be issued to interfaces rather than nodes on the Internet. This scheme will allow a multi-homed host to have a certificate...
issued to each of its interfaces. For instance, a host functions as a home BBR or visitor BBR on different interfaces. It could have a certificate associated with each of these interfaces. In other words, this configuration is particularly useful when the host serves multiple subnets. However we also understand the disadvantage of such a scheme. IP addresses often do not refer to DNS entries directly and therefore one needs to undergo a rather tedious process of a reverse domain name look up to convert an IP address to the domain name. Also one distinguished name may have multiple certificates associated with it further complicating the authentication process. We propose to retain the X.509 distinguished name in the subject name field and include the IP address in the alternate field SubjectAltName. By using this mechanism we distinguish between multiple certificates falling under the same distinguished name.

Similarly the Issuer Name field contains the distinguished name of the issuer whereas the IssuerAltName contains the IP address of the issuer. The Name Constraint field is used in the certificates of BBR and RP and can contain the ranges of IP address corresponding to the networks managed by these entities. For instance, A BBR can specify in its certificate the ranges of IP addresses that fall under its jurisdiction. This can also be interpreted as the range of IP addresses for which a BBR is responsible for generating revocation lists. The Certificate Policies field can be useful for specifying access control related decisions. For instance, using this field one can enforce a guideline that can specify the type of host (e.g., MH or a BR or BBR) and the domains that a node is affiliated with. It can also be used to specify for instance whether a BBR is serving both as a home BBR and a visitor BBR. This information can further aid in making some useful access control decisions. This extension contains a sequence of PolicyInformation objects, each of which consists of an object identifier and one or more optional qualifiers. Because a DLM entity may assume multiple roles, the extension may contain several policy qualifiers, each defining a specific role. As mentioned before, the hybrid certification scheme employs cross certification procedures for two authorities residing in different domains to cross certify each other for administrative reasons. In essence, this can be regarded as a procedure for binding certificate policies of these two domains. Such bindings can be captured in the Policy Map field. Such trust-driven bindings minimises the administrative overheads and at the same time contribute towards providing mobility related services to a roaming host in a transparent way. The Policy Constraints field can be used to enable and disable the policy mapping.

5.6 Summary

The distributed location management scheme proposed in the previous chapter raises several issues relating to security. In this chapter, we have proposed a security model for DLM model. We first discussed key management and public key infrastructure requirements necessary to support our security framework. We then discussed the end-to-end-secure communication protocols between two mobile hosts. We considered four such scenarios. These protocols are responsible for authentication as well as for the establishment of shared session key between two mobile hosts. Finally we discussed extension to X.509 certificates with a view to accommodating certain security requirements pertaining to the DLM model.
Part C

Secure IP Multicasting in Mobile Networks
Chapter 6

Multicasting and Secure Multicasting Schemes for Mobile Networks

6.1 Introduction

A brief introduction to multicasting and traditional multicast protocols such as DVMRP, PIM, MOSPF is covered in Appendix C. These traditional protocols are designed with a static host in mind and did not consider the sources and receivers to be mobile. Hence these protocols are prone to problems in mobile networks. Typical problems in multicasting in mobile networks are [Chin+ 2001]

- Multicast protocols that are based on shortest path trees such as DVMRP may route packets incorrectly or drop packets after MH migration due to reverse path forwarding.
- In shared tree approaches such as PIM-SM [RFC 2362] an algorithm is needed to determine the core called rendezvous point strategic location in the network. This ensures that receivers obtain the least possible delay. In mobile networks, if sources and the receivers are mobile then the RP’s location will be sub-optimal for routing purposes after each receiver/source migration. If the mobile host is able to inform the multicast router about the multicast addresses it wishes to receive, when the mobile host moves again, the datagrams will not be delivered at the new network since the multicast routers have no knowledge about this movement.
- Unless a mobile host is at its home network, multicast datagrams that it wishes to send could possibly be dropped. This is because if downstream-multicast routers receive multicast datagrams on a different interface from that used to send datagrams to the mobile host then the arriving datagram is dropped. Further, downstream routers continue to track reverse paths to the mobile host assuming that the mobile host is on its home network. Thus the multicast routes that are established are always with reference to the mobile host’s home network and are incorrect when the mobile host is at a foreign network.
- At the receiving end, when a MH migrates to a cell with no group members it will experience delay. This is mainly caused by subscription delay, tree rebuild or non-existent multicast routers in the region.
- The Time to Live (TTL) as specified in the multicast packet may be inappropriate. For example, a TTL set for one region may be inappropriate for another. Once the mobile host migrates out of a region, the specified TTL value earlier may be too small.

These issues clearly identify the need for new multicast techniques to be developed and deployed for mobile environments. The IETF Mobile IP [RFC 2002] specifies methods to carry out multicasting in the IETF Mobile IP architecture. Some other proposals in this area of multicasting for mobile computing environment have come from [Acharya+ 96, Wang+ 99, Harrison+ 97, Chikarmane+ 95 and Chin+ 2001]. In [Omar+ 00] several approaches are given to support multicast on IETF mobile IP systems.

This chapter is organised as follows. In section 6.2, we provide an overview of schemes that aim to facilitate multicasting in mobile networks. This discussion
excludes the IETF and Columbia mobile IP, schemes, which, are described in detail in sections 6.4 and 6.5. Securing multicast communication poses several important challenges. The characteristics of a multicast based communication are unique and diverse and it is not always possible to propose a single comprehensive security solution, which best fits such communication models. The notion of mobility further aggravates these problems. In section 6.3 we discuss security issue in the provision of multicasting in mobile networks and then provide an overview of schemes that aim to provide secure multicasting in mobile networks. In sections 6.4 and 6.5, we provide an overview of multicasting schemes in IETF and Columbia Mobile IP proposals schemes and then describe schemes to secure these two proposals. For the IETF scheme, we discuss multicast architecture for Mobile IP in the context of hierarchical local registration model. We provide an overview of the various multicast options suggested by [Omar+ 00] and highlight their relative merits and demerits. We then provide an overview of our secure multicast framework wherein we discuss various issues and scenarios relating to group key management framework and outline key assumptions followed by a description of the security protocols. In particular, we consider the option of securing one of the suggested multicast approaches of [Omar+ 00]. We also discuss the key generation process for creating secure multicast groups along with inter domain authentication and anonymity. In section 6.5, under the Columbia Mobile IP scheme, we describe the approach of MTUNNEL suggested in [Acharya+ 95] and then propose our security extensions to this approach. Finally, in section 6.6 we provide concluding remarks.

6.2 Multicasting Schemes in Mobile Computing: A Survey

In this section, we provide a brief overview of some proposals that aim to provide a multicast facility in mobile networks. We discuss multicasting in IETF Mobile IP scheme and Columbia IP scheme in section

6.2.1 Mobile Multicast (MoM) Protocol

The MoM scheme in [Harrison+ 97] describes a new protocol to support IP multicast for mobile hosts in an IP internetwork. It uses the basic unicast routing capability of IETF Mobile IP as a foundation and leverages existing IP multicast to provide multicast service for mobile hosts as well. The assumptions of this scheme are as follows:

- The service to be provided is unreliable, best effort, connectionless delivery of multicast datagrams.
- Multicast support must conform to host group model of [Deering+ 90].
- A multicast router is co-resident with the HA.
- Foreign agents are being used at the foreign networks.
- Home agents and foreign agents are static and not mobile hosts.

In this scheme, there are no assumptions about the size of the multicast groups, the geographic distribution of the multicast group members, number of mobile hosts in the network, the location of mobile hosts, or the frequency of mobile host movement. The design goals are as follows:
Scalability: The approach should work well even when the number of mobile hosts in the internetwork is large.

Robustness: The disruption of multicast service due to movement of a host from one network to another must be minimal.

Simplicity: The scheme should be as simple as possible, in the sense that it is able to interoperate with existing Internet protocols and mechanisms, with as few changes as possible.

In this scheme it is the responsibility of a multicast HA to forward multicast traffic to the MH through the mobile IP tunnel via the FA. The HA is part of the IP multicast distribution tree rooted at the source. The HA receives multicast datagrams via this multicast distribution tree. The FA need not join groups on behalf of mobile hosts that are members of a multicast group G. Hence this scheme is not susceptible to graft and join delays every time a mobile host moves. The protocol also restricts the setup overhead to the mobile support entities and reduces the impact on internetwork routes not concerned with the mobility support. Therefore, as a result of this approach, frequently moving mobile hosts are expected by the mobility support agents, but not by multicast routers, and the presence of many mobile hosts may overwhelm multicast routers not prepared for the rapidly changing delivery tree.

A HA may be serving MHs at several FAs that wish to receive datagrams addressed to the multicast address for group G. A FA forwards only one copy of the datagram into each mobile IP tunnel. FAs must use link-level multicast at each foreign network to complete the delivery. In case, a MH resides under a FA higher up than its HA in the delivery tree, routing loops may occur as packets from a FA may get re-forwarded down the tree to the HA which re-forwards it back up again. This is fixed by enforcing a time to live of 1 hop on any multicast packet delivered from an FA. To solve the tunnel convergence problem, a selection is performed by the FA to appoint one HA, as the designated multicast service provider, (DMSP), for a given multicast group. DMSP forwards one multicast datagram into the tunnel and other HAs simply suppress multicast delivery down the tree using negative caching as described in PIM Protocol. [Deering+ 94].

The disadvantages associated with this scheme are as follows. When a MH moves from one foreign network to another, it may experience a temporary disruption of multicast delivery service due to the fact that in Mobile IP there is no explicit de-registration with the foreign agent when a host moves. In the case that the moving host’s HA is the DMSP for a group at the previous foreign network, a DMSP handoff will be required to a different HA to forward datagrams for the remaining multicast group members at the foreign network. Until this handoff completes, multicast delivery for group members at the foreign network may be disrupted. This approach also suffers from all the disadvantages associated with home subscription such as sub-optimal routing and possibly long distance tunnelling from home.

6.2.2 Multicasting using AMTree

In [Chin+ 2001], a scheme known as AMTree is proposed that is uses an active network (AN) based multicast tree that is bi-directional, optimisable on demand and
adaptive to source migration. The rationale of the AMTree is to enable the adaptation of multicast tree during handoff using active networks (AN). Current methods are inefficient, require significant computational overheads, or do not consider the source to be mobile. Using AMTree, no modification to the distribution tree is required after handoff. Receivers have the option of optimising their connection to the multicast tree. The reason being after handoff some receivers might experience lower latency while others may experience increased latency. Furthermore, no periodic control messages are used to catch topology changes. This is because receivers are required to join the tree explicitly, thus data only flows over links that lead from the source to receivers. In the case where the link to the parent active router (AR) is down, an optimisation process is executed to graft to another portion of the tree. This scheme does not suffer from traffic bottlenecks encountered in PIM [RFC 2362] and CBT [Deering+ 94] since the tree is basically a source-rooted tree. ARs dynamically filter out unnecessary control messages. For example, join/optimisation/NACK/ACK messages are filtered out by ARs nearest to the subscribing receiver.

In the multicast tree, active routers with at least one receiver/subscriber are termed as core routers. These routers provide an abstraction during handoff. In other words, no modifications are required for routers that are downstream from the core router during handoff. Furthermore, the shortest path is maintained for each receiver connected to the core given that migration in an AMTree is quite similar to that in a CBT. In contrast, the cores in AMTrees are dynamically assigned and receivers do not join using cores as rendezvous points.

Core discovery is performed after handoff and is initiated by the core routers within the multicast tree. The discovery protocol uses the computation at each AR to determine which core/router is nearest to the MH. In order to unicast packets the MH uses the designated core, the core in turn will multicast packets along the tree.

The AMTree protocol is separated into three phases:
- Construction of active multicast tree,
- Update process during migration and
- Tree optimisation by active nodes.

**Building the multicast tree**

This approach is based on the assumption that a distributed location directory (LD) service exists in the AN, which maintains the contact point of a given group, in this case the source. The distribution and access to the LD can be implemented similar to the domain name service (DNS). The construction of the multicast tree comprises three processes: join, leave and send.

**Join Process**

A receiver indicates its interest in a multicast session to its local router. If an active session already exists then no request is made to the LD. Otherwise the local router queries the LD for the contact point. The contact point in this case is the source address or the router local to the MH and it is updated whenever the source migrates. A request message is then sent hop-by-hop towards the contact point. Once an AR that is subscribed to the session intercepts the request message, an acknowledgement is sent back to the initiating router. This completes the join
process. If a receiver is the first to be subscribed then the processing AR loads the AMTree program and creates a state pertaining to the session. The router then is subscribed to the session.

- **Leave process**
  The leave operation in AMTree is explicit. Each subscriber wishing to stop receiving packets sends a leave to its corresponding AR. As a result the subscriber will be pruned from that AR. Apart from that each subscriber maintained at each AR then checks whether it has other subscribers. If no subscribers are found, then a prune message is sent upstream. The upstream AR then removes the downstream AR from its list of subscribers.

- **Send process**: A source interested in sending to the multicast tree indicates its interest in creating/transmitting to a multicast tree through its local router. The local router then registers with the LD stating the current contact point for the session

To summarise, AMTree addresses the three critical issues:

- The foreign network may not support multicast service. Therefore, the receiver is unable to rejoin the multicast session until it migrates to a network that supports multicast.
- The foreign network may support multicast service but does not join the multicast group(s) in which the visiting MH is subscribed to. Hence, the MH has to wait for Internet Group Management Protocol (IGMP) [RFC 1112] next membership query cycle (at most one request per minute) and endure rejoining delay.
- In the case where the foreign network has joined the multicast group, the receiver may receive duplicate packets or subsequent packets. Therefore, the receiver may experience duplicate packets or packet loss.

This approach encompasses the advantages of both source rooted and shared tree with adaptation to host migration. Apart from this, it requires minimal storage and its use of active filtering limits the number of control messages. However the end-to-end latency incurred is only comparable to that of the remote subscription approach of IETF after AMTree optimisation.

### 6.2.3 IP Multicast Support using Mobile Agents

The approach suggested in [Wang+ 99] uses a three-layer architecture for multicast to mobile hosts and introduces multicast agents that serve as the access points of mobile hosts (via foreign agents) to the multicast backbone. In other words, a multicast agent is a multicast router that serves multiple foreign networks. A foreign agent in the service area of a multicast agent notifies the multicast agent of the multicast groups that visiting mobile hosts belong to. The multicast agent joins these groups and tunnels multicast packets for these groups to the foreign agent. At the Internet level, multicast agents are simply multicast routers participating in multicast routing. At the local level, foreign agents still use local multicast as in IP Multicast. However, the intermediate level between multicast agents and foreign agents is responsible for propagating group information of mobile hosts from foreign agents to multicast agents and for tunnelling multicast packets from multicast agents to foreign agents. Routers between the multicast agents and foreign agents do not have to be multicast routers. The use of multicast agents has several advantages.
• Roaming mobile hosts are kept relatively close to their access points to the multicast backbone by restricting the size of the service area of a multicast agent. This also reduces the disruption to multicast services caused by mobility of group members.
• Frequent modifications to multicast trees are reduced since the service area of a multicast agent covers multiple (foreign) networks.
• By ensuring that service areas of different multicast agents do not overlap, the tunnel convergence problem is avoided [Harrison+ 97] where multiple agents may tunnel multicast packets of the same group to the same foreign agent. Only one multicast agent serves each foreign agent.

This approach has the following drawback. Consider the scenario, where a mobile host is the first member of the group in the service area of the multicast agent for the new foreign network. In addition to propagating group membership from the mobile host to the foreign agent and then to the multicast agent, the multicast agent has to join the group through some multicast routing protocol. Due to this disruption of multicast services, the delay can be considerable.

6.2.4 IP Multicast using Mobile IP Tunnels

The scheme proposed by [Chikarmane+ 95] considers the problem of multicast to dynamic groups in the internetwork with mobile hosts. This proposal cleanly separates location management and multicast functions. The assumptions made by this scheme are as follows:

• The disruption of multicast service due to movement of a host from one network to another must be minimised.
• The service to be provided is unreliable, best effort connectionless delivery of multicast datagrams.
• Dynamic group management is a necessary feature of multicast.
• Both static and mobile hosts can be members of multicast groups.
• A mobile host that wishes to receive multicast datagrams is capable of receiving them on its home network without any additional support, using existing multicast techniques.

This scheme is based on an expansion of the virtual network established for IP multicast by using mobile IP tunnels to constitute the “last mile” for delivery of multicast datagrams. This scheme relies on a unicast routing protocol to provide location management and provide multicast as a service accessed from the home network of the mobile host. The goal is to transparently provide multicast capability to mobile hosts in a manner similar in function to that available on the Internet today. The multicast tree is always rooted at the home network of the mobile host. When the mobile host wishes to send datagrams, two scenarios are possible:

• The mobile host is on its home network. It uses link-level multicast scheme to send the datagram. The mHA propagates the multicast downstream normally.
• The mobile host is on a foreign network. It uses a tunnel to deliver the datagram to its mHA. The mHA then propagates the multicast downstream via all virtual
interfaces, including the physical interfaces corresponding to the mobile host’s home network. The recipients of the multicast are unaware that the source is mobile.

In order to deliver multicast datagrams, the HA receives the datagrams via the IP multicast distributed tree set up using DVMRP. The HA then sends a copy of datagram into each corresponding mobile IP tunnel. To solve the tunnel convergence problem a designated multicast service provider (DMSP) is selected for each multicast group. Thus the scheme provides at most once delivery of multicast datagrams, which is identical to the semantics of IP multicast.

This scheme has several useful features. In particular, it provides the ability to support dynamic groups and provide a minimal break in service due to host movement. It also provides an element of robustness due to the availability of alternate multicast service providers in the event of a DMSP crash or a network partition. This scheme also seamlessly integrates with the current multicast protocol specification (DVMRP) and Mobile IP architecture. However, an obvious weakness of this scheme is non–optimal routing that is employed for the delivery of multicast datagrams. Also this scheme relies heavily on the HA for providing the multicast service. The HA can be a single point of failure.

6.2.5 Range Based Mobile Multicast Scheme (RBMoM)

In [Lin+ 2002] a Range Based Mobile Multicast (RBMoM) scheme is proposed. This scheme trades off between the shortest delivery path and the frequency of the multicast tree configuration. In fact, the remote subscription approach and bi-directional tunnelling of [RFC 2002] are extremes of this protocol. Like the home agent in Mobile IP [RFC 2002], RBMoM has a router, called the multicast home agent (MHA) that is responsible for tunnelling datagrams to the foreign agent to which the MH is currently attached. Just as in bi-directional tunnelling, each MHA must always be one of the multicast group members. Every MH can have only one MHA. The HA of a MH is never changed. However, the MHA of a MH is changed according to the MH location. The MHA offers multicast service and the HA offers unicast service. However the initial MHA of the mobile host is its HA. Each MHA is given a range. Both the HA and the MHA need the cooperation of the FA. The range of the MHA means the service range to its MHS. In other words, a MHA can only serve the mobile hosts, which are roaming around the foreign networks, which are within the MHA’s service range, or the network to which the MHA is attached. If the mobile host moves out of a MHA’s range, then a handoff must occur between the new and the previous MHAs.

The MHA information of a mobile host is recorded at its HA. When a MH registers with a FA in a foreign network, the FA contacts the mobile host’s HA to locate the MHA serving the MH. The FA then calculates the distance to the current MHA. If it is greater than the service range, a new MHA must be selected. For simplicity, the authors suggest using the FA as the MHA. If on the other hand, the MH is still in the service range of the MHA, the FA just informs the MHA of the FA currently serving it. In order to get shorter delivery path, if the current FA is already a member of the multicast group, then this FA is made the MHA even though the MH is still in the service range of the current MHA.
This scheme not only offers optimum routes for delivery of multicast datagrams to mobile hosts like remote subscription, it also has less frequent multicast tree construction. The objective is to find the balance of the routing path and the tree maintenance overhead. However, the overhead in this approach is the cost of reconstructing the delivery tree while a handoff occurs. In addition, this approach implicitly assumes that the mobile hosts are only subscribers of a multicast group or they have a collocated address on the foreign network. Also, this approach does not handle the source mobility. The cost of maintenance of MHA and the data structures in HA can be high.

6.3 Secure Multicasting in Mobile Networks: An Overview

6.3.1 Introduction

Most services running on multicast communication infrastructure based on multicast protocols and schemes described above, implicitly assume that the underlying infrastructure provides basic security functionality such that confidentiality and integrity of the information transmitted and authenticity of the group members is assured. For instance, a secure multicast protocol must ensure that only the legitimate members of the group have access to the group information only when they are authorised to do so and that only authorised participants of the group may distribute information to the group. Because secure multicast communications within the group are encrypted, when a new entity joins the group, the key used to encrypt the traffic must be refreshed to prevent the new member from accessing past traffic. Also, there is a need to update the key when a member leaves a group in order to prevent the leaving member from accessing future traffic. This is accomplished using an efficient key management technique in conjunction with various cryptographic techniques.

Various contributions have been made, which propose security related solutions to multicasting in a wired networked environment. The objective of this section is to identify the additional constraints to carrying out multicasting in a mobile IP environment with respect to security and then provide a literature survey of some proposals that aim to accomplish the above.

6.3.2 Secure Multicast and Mobility

Potential security threats to multicast communications are similar to those encountered in unicast communications. Some of the main security concerns in multicast communication as cited in [Shankaran+ 99] are as follows:

- **Secure group communication:** Only legal members of the group should have access to communication related to that group. In other words, non-members must not be able to eavesdrop on multicast traffic. This requires the presence of a confidentiality service for the multicast group. An integrity service is also required to assure that a message has not been illegally altered in transit. The members of the group should be able to detect such illegal modifications.

- **Group and member authentication:** The main issues here are concerned with the authentication of the group and members of the group. These are often intricately linked with key distribution and management as the provision of the authentication service is often based on possession of certain keys. Hence the
problem of securely distributing keys amongst the members of the group becomes a vital concern. It is necessary to provide mechanisms for revoking memberships of those who leave and for registering those who wish to join new. This in turn implies that there must be some secure mechanisms for secure key generation, distribution and revocation as the membership of the group changes. There may also be a need for periodically refreshing the key for the group. Another related issue is the need for individual member authentication in addition to the group authentication. For instance under certain circumstances, it may be necessary to determine which member of the group performed a certain action (eg. sent a message)

- **Multicast group management**: Here the main issues are concerned with who can join the group and who and how one decided who can join the group. This is part of access control. More generally for any operation that changes the group structure, these issues need to be addressed.

- There should also be appropriate mechanisms to counteract the **denial of service attack**. Maintaining service availability against malicious attack is ever more challenging in a multicast setting, as clogging attacks are easier to mount. We do not consider this service availability issue in this thesis.

The introduction of mobility further aggravates these security problems. It is desirable to secure these vulnerabilities in mobile networks while maintaining some of the efficiency and performance benefits of multicast service. Some of the constraints introduced due to mobility are as follows:

- The participating group members have limited computational power. Hence they may not be capable of performing intensive security related computations.
- The group key management process is more complex in the mobile computing environment due to the mobility of the hosts. New keying material must be generated when a mobile host moves from one cell to another or from one foreign network to another.
- Who holds the responsibility for key management functions? Should there be a single static entity responsible for keying material (for example, the home agent in the mobile IP scheme) or should it be dynamic? (Based on the current location of mobile host. For efficiency reasons, it may be necessary for entities such home and foreign agents to become members of the multicast group of behalf of the mobile hosts. In which case, these entities take up the responsibility to process the group multicast information. Since a mobile host has limited resources, it might also need to delegate key and group management functions to these service-providing entities thereby minimising the workload of mobile host. This makes it mandatory for some kind of a trust relationship to be established between these entities and a mobile host.
- Finally, the power, storage and bandwidth limitations restrict the ability of mobile users to participate as peers in the procedures for managing secure multicast groups.

These issues must be given a serious thought when designing secure multicast services in mobile environments for dynamic groups.
6.3.3 Secure Multicasting in Mobile IP Networks: A Survey

Several schemes are in place, which aim to provide secure multicast communication in a mobile computing environment. In this section, we provide an overview of these schemes and discuss their relative merits and demerits.

6.3.3.1 Secure Multicast in Wireless Networks of Mobile Hosts

This scheme is described in [Bruschi+ 00]. This scheme considers a cellular architecture for mobile environments consisting of two sets of entities: a set of mobile hosts, and a set of static, or fixed hosts. A wired communication network connects all fixed hosts. Fixed hosts with a wireless network interface are called Mobile Support Stations (MSSes) as they communicate directly with the mobile host and provide the wired infrastructure. Three different multicast scenarios are identified based on the degree of trust in MSSes. These scenarios fully characterize the various configurations where a secure multicast protocol may be implemented in a cellular wireless network. The three scenarios are:

- **Non-trusted system** consisting of cellular networks with possibly malicious MSSes, i.e., MSSes that might try to gain access to data traffic within the group, even if they are not allowed to.
- **Semi-trusted system**, consisting of cellular networks with MSSes that are not allowed to understand the data traffic within a group but behave correctly with respect to a given protocol.
- **Fully trusted system** consisting of cellular networks with trusted MSSes, i.e., MSSes that may have access to traffic that will never reveal confidential information related to the group.

For each of these scenarios [Bruschi+ 00] propose a key management protocol upon which a secure multicast service can be built. These protocols balance two conflicting requirements; namely, limit the group manager and the MSSes load, so as to avoid multicast implosion, while maintaining the computational overhead of the mobile hosts as low as possible. The separation of group dynamics and host mobility management tasks among the three components of the systems, i.e., the group manager, the MSSes, and the mobile hosts is the key idea in the design of these protocols.

For non-trusted systems the group manager and mobile hosts that are members of the group share the task of managing the group. It may be argued that in this case it will be better to adopt an efficient multicast protocol selected from among those proposed for wired network such as [Wallner+ 99] and [RFC 1949]. The reason being that there is no trust to trade for efficiency. The selection of the protocol can be determined by the way the load is distributed among the participants. One option is that the group manager takes all the key management aspects. This approach limits scalability and the group manager becomes a single point of failure. In the other case, the mobile hosts collaborate with each other and therefore perform considerably high percentage of work. This may be unacceptable as due to certain constraints that were highlighted above.

For semi-trusted systems, a protocol is suggested that balances the load of a secure multicast primitive among system components. This has the disadvantage of not being
an optimal scenario for a mobile host, since it is burdened with a fixed overhead due to additional cryptographic operations that allow the group manager and MSSes to limit their management of host mobility.

For a fully trusted system, a protocol is proposed that relieves the mobile hosts of the additional cryptographic operations and minimises their participation in the key management process. Mobile hosts are required to perform data traffic decryption only. In this case the overhead of group dynamics and host mobility management is completely on the group manager and the support stations. The main drawback of this protocol in the semi-trusted approach is the fixed overhead of double decryption operations each mobile host must perform in order to access the key used to encrypt the data traffic. Such a drawback is overcome by the protocol for fully trusted systems where only the mobile hosts decrypt the data traffic. The greater costs of this solution to the previous one are compensated by a reduced overhead of the mobile hosts. From the security perspective, the advantage of using the non-trusted support stations is that their security is not a concern for the correct execution of the protocol. Furthermore, the separation between the security and networking aspects is clear. On the other hand, as the degree of trust in the support stations increases, thus increasing their involvement in the key management protocol, their security becomes critical for the correct execution of the protocols. In this case, security and networking aspects are not as clearly separable as before.

6.3.3.2 Multicast Security Extension to Mobile Environment

This proposal is described in [Gong+ 95]. This proposal considers two cases:

- Case One: Accommodating a single mobile host.
- Case Two: Incorporating a group of hosts that participate in a multicast session through a shared network.

In case one, in order to connect to a multicast distribution tree, the mobile host initially uses a host in its home network. This is a full trust agent that distributes certificates and keys. This trusted home host participates in the secure multicast session and forwards the data to and from mobile host via the tunnel they established. Once the mobile host is well established in the session, it takes the responsibility for data privacy and assigns another agent to forward the multicast messages along a shorter path to its home agent. This agent operates under partial trust and carries out on behalf of the mobile host only session management tasks. However, group encryption keys are held and used by the mobile host and are not known to the agent. In other words, this foreign agent forwards encrypted data only.

A selected agent can be a host dedicated to this function; or a host can become an agent upon request. In a relatively static setting where the users are staff of a corporation, and using machines in their offices, an agent can be a server on a local area network. In a more mobile setting, say, when a travelling executive takes part in a session form a hotel room or some other public connection point, an agent is typically located within the executive’s laptop or hand held device, which in this case acts as a fully functional node on the M-Bone. It should be noted that the mobile computing device need not retain its home IP address, and may not need an in-case care of address either, as long as it can participate in multicast and obtain access to
session information. Alternatively, the user must locate an agent somewhere in the network, and can use point-to-point messages to inquire about a suitable agent nearby.

A sender in a multicast session transmits a message to its secure multicast agent, which may then choose to encrypt the traffic with a suitable session key before handing over the message to the (untrusted) multicast mechanism. At the receiving end, a client obtains information through subscription to its agent. A session participant’s agent examines all incoming multicast traffic to see if anything is of interest to the participant, and forwards information as appropriate.

The advantages of an agent-based approach are simplicity and backward compatibility. For example, security issues are separated from transport and routing issues. Also the burden of providing security is on the mobile hosts and the security agents, and not on disinterested machines and routers. To reduce bandwidth consumption, the agent can perform most of the data filtering and organization work for the mobile host. This approach can also be IP multicast based and should not require modifications to IP-multicast or lower layers of the network architecture. The disadvantage of this approach is that if a mobile host cannot find a multicast agent it must perform complicated key distribution functions on its own. This proposal does not specify secure handoff procedures to a new agent upon mobile host’s change of location.

This agent-based architecture is extended to accommodate a group of mobile hosts, which share a common broadcast channel while participating in a multicast session. This is accomplished by the mobile hosts agreeing on a common agent and a group key and the broadcast of the information to the group using that key. There are some drawbacks with this approach. The problem with this approach comes when the agent selected is fully trusted. In this case, the process involves the transfer of key management for each host’s agent to the selected agent. The process is cumbersome and time consuming. There are also no solutions when the mobile hosts may require agent assistance in their security functions and yet cannot agree on a single trusted agent for all their traffic. The authors allude to developing a hybrid solution to cater to this requirement.

6.4 Multicast Extensions to IETF Mobile IP

6.4.1 Multicast in IETF Mobile IP

There are two approaches proposed in [RFC 2002] to carry out multicast in mobile IP. These approaches are bi-directional tunnelling and remote subscription respectively.

In bi-directional tunnelling approach, the home agent (HA) becomes the sender and receiver for multicast traffic on behalf of its mobile host (MH). Whenever the mobile host migrates to a new subnet, a bi-directional tunnel is created from the mobile host’s care of address to the HA. As a result any traffic generated by the MH or directed towards the MH has to traverse through the HA. MH when tunnelling a multicast datagram to its home agent is required to use its home address as the source IP address in the inner multicast datagram. This second method is used only when the home agent is the multicast router. This approach handles source mobility as well as
recipient mobility, and in fact hides host mobility form all other members of the group.

There are several drawbacks with this approach. First, this scheme does not facilitate optimal multicast routing. In the worst case, the source and the recipient can be on the same network, while all the multicast messages between the two hosts must traverse the entire internetwork twice. Second this method suffers from tunnel convergence problem [Harrison+ 97]. Imagine a situation when mobile nodes, belonging to different HAs move to the same FA. Using bi-directional tunnelling approach, each of the respective HAs creates a separate bi-directional tunnel to the FA so that multicast packets can be forwarded to their respective MHs. If these MHs are subscribed to the same group, all the tunnels from different HAs to the FA start carrying the same multicast packet and this results in packet duplication. This is known as the tunnel convergence problem.

In the remote subscription based approach, a care of address is allocated to the MH when it moves to a foreign network. The care of address is then used to provide multicast. The mobile host joins the group via a local multicast router present on the foreign network. This option assumes that there is a multicast router present on the foreign network. A Mobile host uses the co-located care of address as the source IP address of the IGMP [RFC1112] messages; otherwise, it is required to use its home address. This method is simple and works well if the mobile host spends a relatively long time at each foreign network compared to the join and graft latencies in other approaches such as [Acharya+ 96].

This scheme offers the advantage that datagrams are delivered via the shortest path. There are several drawbacks too. The approach implicitly assumes that mobile hosts are only recipients of multicast messages or that they have a co-located care of address on the foreign network. If the mobile host sends a multicast datagram with its home address as the source, the incoming interface check as in [RFC 1085] or [Deering+ 94] may discard datagrams intended for the multicast group. This approach also assumes the existence of a multicast router at the visited network, an assumption that may not always hold in an IP internetwork. Without such a multicast router, multicast message delivery can be achieved only by using some form of tunnelling. Another disadvantage surfaces with the use of source routing trees. This tree needs to be rebuilt every time the mobile host moves to a new location. Although the end-to-end latency is lower, the handoff latency may be much higher because of the time taken to rebuild the tree. Due to mobility, packets directed towards the mobile host might not be forwarded correctly and reconstruction of the tree each time the mobile host migrates makes this scheme inefficient.

The problem with the base Mobile IP protocol is that every time a mobile node moves from one network to another it needs to register its new location with its home agent (HA). Transmitting and processing all the registration requests through the remotely located HA may become inefficient. The base Mobile IP protocol architecture was extended to decrease this overhead. This extension is called the hierarchical local registration mobile IP [Perkins 96].

In this scheme, the FAs are arranged hierarchically in the regional topology and the MH is allowed to move from one local area to another of the same topology without
the need to send a registration request to the HA (refer to figure 11). The MH only informs the corresponding local FA each time it moves to a new area. This approach eliminates the registration delay involved in contacting the HA each time the MH changes its location. Consider Figure 1. The system has one HA and two root FAs FA$_1$ and FA$_2$. The nodes named FAx are FAs supporting local registration, while nodes labelled Rx are regular routers supporting no FA functionality. Each FA announces the higher part of the hierarchy that this FA is located on it. For example FA$_4$ will announce the chain FA$_4$/FA$_3$/FA$_1$ while FA$_8$ will announce FA$_8$/FA$_6$/FA$_3$/FA$_1$. It is only when the MH moves to FA$_{21}$ (step 3) then the MH needs to send the registration request to the HA. As long as the MH is moving within the area served by the same root FA, only local registration is needed and the HA need not be involved in the MH registration. When the MH moves beyond the scope or hierarchy of the current root FA, the MH will need to send the registration to its HA to inform it about the new root FA.

6.4.1.1 Supporting Multicast in Hierarchical Mobile IP

The IETF has proposed two approaches [Perkins 96] to provide multicast support. The first approach utilizes the current FA serving the MH to join the multicast group while the HA is considered by the second approach for the same purpose. In order to support internetworking with existing multicast infrastructure, the mobile IP system using either FA or a HA subscription method must support the underlying multicast routing mechanisms.

In the FA subscription approach, the MH will notify the FA about the multicast group for which it wishes to receive datagrams. If the FA is a part of this group then the MH will start receiving datagrams straightaway. Or else the FA will use the available multicast protocol to become a member of this group.

The HA subscription option utilizes the technique of bi-directional tunnel. A MH will establish a bi-directional tunnel to FA. The membership reports generated by the MH will be sent to the HA via this bi-directional tunnel and thereafter the HA will register with the multicast group on behalf of the MH. The HA will use this tunnel to redirect all multicast datagrams to the mobile host.

In [Omar+ 00] issues associated with each of these approaches are described in detail along with different schemes that efficiently support multicast in hierarchical local registration environment and take advantage of the inherent characteristics of hierarchical systems. In this chapter we will describe them in some detail.
There are two proposals described in [Omar+ 00] for extending mobile IP to support multicasting. The first approach is through FA in the foreign network and the second approach is using the home agent of the mobile host. We shall describe each of these cases in some detail.

**FA subscription in Local registration System**

There are two schemes in this approach: Root FA Subscription and Intermediate FA subscription. We will examine them in some detail.

**Root FA Subscription**

In this approach, the root FA must join the multicast groups of interest to MHs, which reside in the lower levels of its hierarchy. The lower FAs discharge group summary reports by means of which it keeps its group information up to date.
A MH wishing to join a multicast group sends a membership request to its (current) FA. The FA examines its database to determine if it is already a member of this group and has been receiving packets for this group. If it is not, it must send a summarized report to its parent FA (next level up). This summarized report contains multiple entries one for each requested group, covering all groups requested by the MHs residing within the FA’s service area. No report is to be generated if the FA determined that it is already receiving the multicast traffic for the requested group. The summarized report will then be forwarded along the hierarchy towards the root FA.

When an intermediate FA receives multiple reports from different FAs in the lower hierarchical level regarding the same multicast group, or when the report is concerning a group that the FA has already subscribed in, the FA will update, summarize or suppress the forwarded report to the upper level while updating the entries in its local database if needed. On receiving this membership report, each intermediate FA will intercept the report and create an entry indicating the interest of the lower FAs in receiving the multicast traffic associated with specific groups. This process will result in that the root FA will receive from each of the FAs in the level below a single report listing the groups that the FAs need to receive traffic from. Accordingly the root FA will join all those multicast groups using the underlying IP multicast infrastructure.

Upon receiving a multicast datagram a FA will consult its local database to determine if it has any MHs belong to this group. It also determines if any lower level FA has subscribed to this multicast group. These steps get repeated at each level till the last FA that is interested in this group receives the multicast datagram.

When a MH moves from one serving area to another, in addition to sending a local registration request it sends a membership report to the current FA to be forwarded to a common ancestor FA (of both the previous and current FA). This common ancestor FA need not forward the report, it only needs to update its entry to point to the FA in the lower level on the new lineage instead of that of old one. The simplest mechanism to remove multicast entries from the old FA is to rely on next membership report from the previous FA that carries the most updated information. This comes with the cost that multicast traffic will continue to be delivered unnecessarily for some time to the old FA. If bandwidth is scarce then the common FA may send a query for group membership in response to a local registration request query. The common FA extracts information on the old FA and sends a membership query to this FA and this will cause the old FA to respond back with its report. Since this report is sent upward towards the common FA, it causes the intermediate FAs to incorporate their membership into the summarized report. The common FA will then send a single summary report from the lineage associated with the old FA.

Let us examine some of the issues that need to be taken into consideration with this approach.

**No Optimised Routing Procedures:** Multicast destined for different FAs has to flow in the hierarchy passing through the root and the intermediate FAs. This approach does not provide an optimum route in contrast to the one resulting if the destination FA joined the multicast tree directly.
If a MH forwards a request to join a group G and its current FA is not a member of this group then the request moves upwards. If the MH is currently in a sub hierarchy that is totally disjoint from its previous sub hierarchy, then one does not encounter any common ancestor that is receiving packets for this group. In which case, the request arrives at the root, which processes this join request. If the group members are sparsely distributed, it may well happen that several join messages get propagated to the root for different groups. The root becomes a bottleneck in this case.

**Root becomes a single point of failure:** In this scheme, the root FA is the only entity that must join the multicast tree, in which case, it becomes a single point of failure.

**Intermediate FA Subscription**

This approach eliminates the above-mentioned problems but comes at the cost of requiring more FAs to join the multicast trees. To reduce the overhead of associated group join/leave this approach intelligently distributes the burden amongst different strategically selected FAs called the Local Multicast Service Providers (LMSP). The LMSP basically provides multicast service for the FAs in its domain, which includes those FAs on lower levels not serviced by other LMSPs.

The LMSP selection can be based on either topology or on MH density. In the case of topology, each hierarchy is logically divided into multiple segments. A FA is assigned for each segment to be the LMSP and is responsible for joining the multicast groups requested by the FAs associated with the particular segment. This scheme may be used to avoid adding excessive delay to datagrams received by the FAs on lower levels of the hierarchy by selecting segments and FAs fairly distributed over the hierarchy.

In the case of the MH density approach, the hierarchy is logically split into multiple segments with each segment having its own LMSP. The FAs at the bottom of the segment send their group report inwards including the count of MHs interested in a particular multicast group. Each FA keeps a track of the number of MHs, and as soon as the number associated with a multicast group exceeds a threshold then the FA will position itself as the LMSP.

Unlike the root FA approach, here the group membership report generated in this approach contains an additional entry that records the total number of MHs wishing to receive traffic of different multicast groups. The LMSP assumes the responsibility of receiving and forwarding datagrams downwards after consulting its table. Eventually datagrams will not be forwarded anymore when received by the last interested FA in the domain. Within a segment, a datagram is replicated the minimum possible number of times.

A movement of a MH can be classified as intra-domain or inter-domain. In contrast to the root FA approach, where the common ancestor FA between the old and the new FAs in the same hierarchy is guaranteed to have an entry for the multicast group requested by the MH, inter-domain movement may necessitate the new LMSP to join the requested multicast group. A procedure to remove old entries and pointers similar to that mentioned in the root FA subscription approach will work in this case also.
Issues

If a MH becomes a member of the group G and its current FA does not reside in the same segment as the current LMSP of the group then this may result in routing the packets via the root down to the concerned LMSP. An increase in the number of MHs belonging to a group G in such disjoint hierarchies cause an increase in traffic at the root.

HA Subscription

In this scheme, the HA will join the delivery tree associated with the multicast group requested by its MH. The MH sends its unicast membership to its current FA, which then forwards this report along with the details of the HA supporting this MH upwards to the root. It is expected that the root FA will receive multiple reports from different FAs requesting to receive the multicast traffic associated with the same multicast group through different HAs. The root FA considers those HAs as HA-MSP candidates for this group and selects only one to be responsible for forwarding the multicast traffic.

Different scenarios have been illustrated for the selection of the HA. The datagrams forwarded from the HA will be encapsulated and sent to the MH, then re-encapsulated again with the destination of the root FA. The HA will send one copy of the multicast datagram to the root FA even if there is more than one MH currently have local binding for this root FA. On receiving the forwarded multicast packet, the root FA will de-capitalize the datagram, identify the multicast group and forward the datagram to the FAs. In the lower hierarchical level as indicated in this local table. The destination FAs will use link level multicast to forward datagrams to recipients.

In this approach, multicast datagrams are forwarded over the same route as that of unicast traffic. The standard local registration scheme requires the MH to send a local registration request when moving to a new area within the same hierarchy. Upon moving to a new service area, a MH has to send a new membership report to the new FA. The mobility of the MH affects the system in the case when the MH is moving into an area served by a new root FA. And it happened that this MH is the only host that the HA-MSP has binding for this group. In this case the root FA has to select a new HA-MSP.

Issues

This approach leads to the triangle routing problem that results in a less than optimum path and therefore latency is high. There is also the problem of large number of undelivered datagrams during HA-MSP switching.

6.4.2 Secure Multicast Framework: An Overview

Securing multicast (group) communication is fundamentally different from securing unicast (point-to-point) paired communication. In this section we outline a secure multicast architecture for the multicast approaches proposed in [Omar+ 00]. The proposed architecture includes secure group membership management and key management facilities for groups of principals that may use multicast
communications. It also includes the interaction between various principals using secure protocols. Any solution proposed must take into consideration various distinct characteristics such as group size, membership dynamics, topology, and degree of interaction, latency requirements, centralized control, and bandwidth constraints.

6.4.2.1 Assumptions

Our architecture is based on the following assumptions:

- A public key infrastructure is in place in the form of a certification authority (CA) or a hierarchy of certification authorities for the purpose of authentication and public key distribution. A certification authority is a trusted entity that verifies the identity of a participating entity, allocates a distinguished name to it and vouches for the identity by signing a public key certificate for that entity using a private key.
- Every host is initially registered in the network that acts as its home. It receives a certificate that is signed by some CA that is local to this host.
- Each FA has a public key and maintains a cache of the public keys of its immediate peers, its subordinates lower down in the hierarchy, its parent as well as the root FA.
- Each MH has its public and private key pair along with the public key of the current FA and in some cases common ancestral FA.
- The root FA has a public key and maintains a cache of the public keys of its immediate peers and its subordinates lower down in the hierarchy.
- Every HA has its public key along with the public keys of all FAs with which it currently has associations.
- All FAs share a group key with their parent FAs as well as with their Peers.
- FAs with directly connected MHs maintain a secret key with each MH. They also maintain a cache of public keys of all MHs currently under their jurisdiction.
- Each FA is trusted for authenticating a mobile user using public key information and in some situations for maintaining the relevant multicast group information.
- If an FA is found to be non-operational, it may be necessary for its parent FA to recompute the group key.
- Each multicast group has a designated member whose responsibility it is to determine who may belong to belong to this group. Normally, this entity is the initiator of the group. If the initiator leaves the group, a new designated member can be chosen via an election algorithm or some other equivalent manner. In some cases it may well be that this responsibility is performed by the FA itself on behalf of the initiator. We develop our model based on the assumption that FA acts as a group manager.
- The beacon message generated by an FA includes its public key certificate.
- Our security model is based on IGMP v2 [Fenner 97].

In this chapter, we will consider security for the intermediate FA registration approach. Before proposing a security model for this approach, we will first discuss the requirements of a secure system in the hierarchical registration approach and thereafter suggest a security solution for the intermediate FA registration approach.
6.4.2.2 Group Key Management Issues

A mobile host (MH) wishing to initiate a multicast group creates an access control list (ACL) for the group. It then announces this group by sending a unicast message to its current FA along with the access control list associated with this group. The current FA retransmits this information upwards towards the root. All intermediate FAs as well as the root may cache this information.

The key issue here is who performs the role of the group manager, which is a trusted entity that is responsible for the generation and distribution of keys. The following are some options that are available for the schemes outlined above:

- The trusted entity can be the group initiator (some MH that wishes to start a group).
- The trusted entity is the Home Agent associated with MH.
- The trusted entity is the FA under whom MH first initiated the group.
- This trusted entity is a FA where MH is currently located.
- There is no static FA acting as a trusted entity. The trusted entity gets replaced using the push approach depending upon group membership distribution.
- The Trusted entity is the root FA. The root FA takes the responsibility of key generation, distribution and updation. Lets examine each of these scenarios.

Case One: MH is the initiator of the group and also acts as the Group Manager

The entity that initiates a group in the first instance is the mobile host. Hence it is the natural choice for this node to take the role of group manager for groups that it has created. But there are some issues here. The MHs normally have limited storage capacity and may be powered down frequently. They may also often go into the doze mode to conserve power and may not be available on line. The process of key generation and storage may prove to be too cumbersome for a MH to manage. This increase in overhead is directly proportional to the increase in group size, frequent changes in membership and the number of groups supported by the MH. Also the entity that performs these activities must always be available on line, a requirement that MH may not always be able to satisfy.

Case Two: Home Agent of MH acts as a Group Key Manager

In this case, the key generation, distribution and updation becomes the responsibility of the HA of the mobile host. Since by default, the HA manages the location details of a MH, it may seem to be the right choice to carry out the group management functions. Furthermore, it is natural to trust a single static entity (HA) rather than frequently changing alien entities (FA). Therefore a scheme based on the HA approach offers high reliability. This approach also eliminates the constraints of limited storage and disconnection problems as in case one.

However, this approach introduces latency as every join and leave must be forwarded to the HA and processed by the HA. It may take an unacceptably large amount of time for a leave message to get to HA and for the HA to update and redistribute the keys. In this transit period the MH may continue to receive packets for the group. This
approach introduces a less than optimum delivery path in the form of triangle routing. The HA also becomes a single point of failure.

Case Three: The current FA of MH acts as the group manager

This approach, just as with case two, eliminates the problems of limited storage and disconnection of case one. If the group is initiated under a foreign agent (the current foreign agent) then that foreign agent takes the responsibility of group manager. We shall call this FA the GIFA. This GIFA then sends a notification message upwards towards the root with information related to the group. This information can be simply a tuple \([\text{Groupid, Group manager (FA)}]\). The root FA as well as the intermediate FAs cache this information. In this case, the GIFA performs all the group key management functions. Even if the group initiator MH leaves the group or moves to a new location (under a different FA), the GIFA continues to perform this responsibility till the group becomes non-existent (no members are registered with this group). Any join request generated at any portion of the subtree needs to be processed by this foreign agent before allowing any node to be a member of the group. In this case, the latency for join is determined by the current location of the MH (how far away is it from the concerned FA) and the numbers of requests being handled by this FA.

The problem with this approach is to locate the group manager for a particular group. One approach would be for GIFA to simply broadcast this information to all other FAs using flooding as a simple technique. The amount of traffic generated will be directly proportional to the number of new groups that come into existence and this may strain the available bandwidth considerably. To solve this problem, we extend this architecture by utilizing the services of the root FA. The root FA plays the role of the Central Location Server (CLS) as it has the list of all currently active groups as well as their group manager. Every GIFA must register itself along with the groupids that it supports with the root FA. If the group ceases to exist then GIFA must de-register the group with the root FA. Other intermediate FAs act as Partial Location Servers (PLSs) as they cache only group related information, which are propagated along their subtree from the leaf towards the root and update their caches by processing register and de-register messages from the node to the root.

Consider the following scenario. Assume that a MH with id MH\(_1\) initiated a group under a foreign agent FA\(_1\). This FA\(_1\) has assumed the responsibility of the group manager (GIFA\(_1\)). The FA\(_1\) then sends the tuple information upwards towards the root with all intermediate FAs as well as the root caching this information. A MH with id MH\(_2\) under a foreign agent FA\(_2\) (which does not lie along the hierarchy tree of FA\(_1\) and the root) initially sends a request to join group G to FA\(_2\). The FA\(_2\) upon consulting its table realises that it is not receiving any packets for this group G and therefore has no members belonging to this group G. It then generates a request, which is propagated upwards along its hierarchy subtree towards the root. In this case, the request of FA\(_2\) is received by the CLS, which supplies the location details, which propagates down the subtree tree towards FA\(_2\). This information is the tuple \((G, \text{FA}\(_1\))\), which is cached by all intermediate FAs as well as FA\(_2\). FA\(_2\) then uses the regular routing infrastructure to route its request to FA\(_1\) (i.e., the best optimum path available). FA\(_1\) receives the request, verifies the ACL to determine if MH\(_2\) meets the required criteria to join the group and then responds back to the originating FA FA\(_2\). The response can be an Accept or a Deny message. If the request sent by FA\(_2\) is
accepted then FA₁ includes FA₂ in its group table. The group table is referred to by FA₁ to route multicast packets. The packets are then routed as dictated by the underlying multicast routing protocol deployed by the network. This approach facilitates the use of best optimum path available and also avoids the problem of single point of failure, as location information is not only held at the CLS but also by the intermediate FAs in the subtree.

**Case Four: Always the Current FA of MH (initiator of the group) acts as the group manager.**

Assume that a mobile node MH₁ initiates a group G under a foreign agent FA₁. The FA₁ propagates this information upwards in the form of a tuple as in case two. If MH₁ changes its location and moves under a new foreign agent FA₂, it requests FA₂ to act as the group controller. In this case, FA₂ requests FA₁ to transfer the group state information to FA₂. FA₂ then releases the tuple information upwards along its subtree towards the root. FA₁ does this too which indicates that it is no longer acting as a group manager for this group. The tuple is of the form \((G, FA₁, \text{Delete})\). This is necessary as all intermediate FAs between FA₁ and the root must update their caches. This approach is useful in situations where the members of a group are tightly coupled with the group initiator node i.e., they migrate together as the initiator of the group moves. The overhead associated with the routing is eliminated.

If the MHs are highly mobile then there will be frequent handoffs between the foreign agents. This problem can be mitigated if some ancestral FA higher up in the hierarchy manages the group membership instead of the leaf FAs. If the group members are restricted to one particular FA then this FA seems to be the ideal choice to carry out group management functions. Note that this architecture is extended wherein the root FA is utilized as a location server. In this case, when the new FA takes charge of the group it notifies the root. The root then updates its table by deleting the binding associated with the previous FA for this group.

**Case Five: Push Approach: Responsibility for group management moves upwards from the current leaf FA towards the root.**

In this approach, it is assumed that group members are distributed such that they do not have a common ancestor. The solution is to push the key generation responsibility upwards to a common FA, which is a parent to all lower level FAs that contain members of this group. This approach, known as the push approach, incrementally pushes the group key information upwards towards the root based on the mobility patterns of member MHs. It becomes simpler if all the members of a group reside under a single FA (the leaf FA) in which case this current FA acts as the group key manager as in case four. In the intermediate FA approach the push may halt at LMSP.

**Case Six: Root FA acts as the group manager**

The root FA becomes a member of all currently active groups and therefore assumes the responsibility of group manager. The MHs often change FAs frequently but rarely change their root FAs. Therefore, instead of pushing the key management information incrementally upwards, all group-keying functions are handled automatically by the root. Since the group membership information is maintained only at the root, this
approach is relatively static and eliminates the overhead associated with handoffs as in cases four and five. However, whenever an MH joins or leaves a group, the update notification messages must be relayed all the way to the root, which then has to refresh the keying information and then propagate it downwards. All requests to join and leave the group are processed centrally at the root. Since the root supports all the currently active groups, it may be inundated with several group management related information and hence may become a bottleneck and a single point of failure. This approach also introduces the use of less than optimum paths to deliver packets thereby increasing latency.

6.4.2.3 Observations

Upon examining these cases in some detail, it is clear that the push approach has the advantage of eliminating the single point of failure problem as in case five by distributing group management functions fairly within the hierarchy. It also eliminates the need to change the group manager with a change in location of the MHs as in case three. It simplifies the role of the leaf FA by taking the group management responsibilities away from them and thereby eliminates the overheads encountered by leaf FAs as in case two. The push approach also experiences minimum latency in join and leave when compared to the other approaches.

The architecture can be further simplified by making the root FA act as a location server. Once a FA declares itself as an LMSP for a segment (based on either topology or MH density) it registers this information with the Root FA. If a group is freshly initiated by a mobile node MH and there exists an LMSP in its hierarchical tree then this LMSP by default becomes the group manager for the group initiated by the MH. During its registration phase, it supplies the following information to the root:

(FAid,-LMSP, G_list, G_IList).
FAid-LMSP: Binding its identity with the LMSP tag. In other words it declares itself as an LMSP.
G_list (Group List): The group for which it currently receives packets.
G_IList (Group Initiator List): The groups for which it acts as a Group manager.

When a MH desires to be part of a group, it sends a request to its current LMSP. The group manager LMSP (gmLMSP) of this group will decide if this MH can be a part of the group. How does the current LMSP of the requesting MH determine the gmLMSP of the concerned group? There are two options here, the push approach or the pull approach.

In the push approach, every time a LMSP is registered with the root and if this LMSP is a group manager for any group then the root propagates this information to all the other LMSPs. This way the root need not be bothered processing group manager queries. All the LMSPs have updated information about each other’s role as group managers. However, this information is unnecessarily replicated at several nodes (depending on how many LMSPs are currently active).

In the pull approach, the root simply caches this information. When an LMSP sends out a request the root supplies the corresponding information. There is a single repository and no duplication of effort is introduced. However, the root stands as a
single point of failure. To eliminate this problem all intermediate FAs along the sub-tree of the concerned LMSP can cache the response generated by the root.

6.4.2.4 Secure End-to-End Multicast Protocols

In this section we propose a set of secure end-to-end protocols for multicast communications based on the assumptions and issues discussed above. In particular, we focus here on securing case five using the LMSP approach for reasons as outlined in the previous section.

Assumptions for the Intermediate FA Approach Model

Some security related assumptions specific for the intermediate FA registration approach are as follows:

- All lower level FAs under an LMSP cache the certificate of the LMSP and vice-versa.
- The root caches all the certificates of currently active LMSPs.
- All LMSPs cache the certificate of the root.
- An LMSP that acts as a group manager for a group G is known as gmLMSP.
- A gmLMSP for a group G must cache all the keys of all mobile hosts and other LMSPs that are members of this group. A gmLMSP also performs the admission control functions related to the group.
- All FAs receiving packets below a LMSP need not be visible to the gmLMSP. In other words, the local LMSP keeps track of group membership for all lower level FAs and becomes a member of a group on behalf of all lower level FAs.
- If a gmLMSP observes that there are no group members currently active for a group for a lifetime $T_1$ (as set by the system personnel), it registers the group to be inactive. The inactive period is a waiting period before the group related information is removed. At the end of the inactive period, the gmLMSP declares the group defunct by issuing a group dead message to the root. The root propagates this information to all other LMSPs.

Case One: Group Subscription: MH joining an existing multicast group

Consider the situation where a mobile host MH wishes to join an existing multicast group G. Let us also assume that the current LMSP of the MH is the group key manager for this group G. In this case, there are two levels associated with the group join process. Assume that a mobile host MH served by a foreign agent FA$_{11}$ wishes to join the group. Also let us assume that this FA$_{11}$ is currently not receiving any packets destined for this group G. In order for MH to join and receive packets for this group G, FA$_{11}$ must first register with the GMLMSP of this group. An FA will join a group only if it has at least one MH under its jurisdiction that wishes to participate in this group. The FA and the MH must first authenticate to the LMSP that acts as a group manager for this group. However, an additional security check for the MH, not applicable to the FA is that the MH must fulfill the requirements of the local group policy as determined by the access control list for that group. Once the authentication of the concerned MH is successful then its current FA by default becomes a member of the group and receives packets for that group on behalf of the mobile host. If the MH is denied access to participate in the group and the FA serving this host has no other members interested in this group, then this FA by default does not join the
group. There is a lifetime associated with the group membership. The lifetime of an FA membership is long lived whereas that of a MH is short lived. Each member must periodically refresh its membership when it is close to the end of its lifetime.

The process of registering with a multicast group is as follows. The MH first obtains the public key of the receiving FA_{11} (FA_{11}) from the certificate in the beacon message that the FA_{11} sends. We shall assume that this FA is already receiving packets for this group and therefore has a lifetime associated with its membership. The MH then sends a join message as part of the membership report. This join message is typically an IGMP v2 protocol message [Fenner 97] that is extended it to include security-related information as follows:

(1) \( MH \rightarrow FA_{11}: CERT\-MH, MH, T_1, N_1, FA_{11}, G, \{T_1, N_1, MH, FA_{11}, G\} SK-MH. \)

Where

- MH is the identity of the mobile host that wishes to join the multicast group.
- CERT-MH is the public key certificate of the mobile host
- FA_{11} is the identity of the receiving HA
- T, N: Time stamp and Nonce generated by MH
- \( \{...\} SK-MH: \) This notation implies that the contents within \( \{...\} \) are hashed and signed using the private key of MH, SK-MH. In this case, it includes the nonce, timestamp, identity of the FA_{11}, and the address of the multicast group (G) that MH wishes to join. The signed element is referred to as the token of MH.

The message contains a signed time stamp T and nonce N to prove its freshness and to protect against replay attacks, the signed element also includes the identity of the MH and FA_{11} along with the group G that the mobile host wishes to join.

Upon receiving this message, FA_{11} verifies the certificate and uses the public key of the MH recovered from the certificate to verify the signed element. It checks the integrity of the message. It also checks the time stamp and nonce to establish whether the message is fresh.

It then constructs the following message that is dispatched to the LMSP of its subsegment:

(2) \( FA_{11} \rightarrow LMSP_1: CERT-FA_{11}, FA_{11}, LMSP_1, T_2, N_2, MH, CERT-MH, \{LMSP, FA_{11}, T_2, N_2, MH, G \}SK-FA_{11} \{T_1, N_1, MH, FA_{11}, G\} SK-MH. \)

Where

- CERT-FA_{11} is the certificate of the sending Foreign Agent FA_{11}
- LMSP_1: Identity of the LMSP of this segment
- T_2, N_2: Time stamp and Nonce generated by FA_{11}
- CERT-MH: Certificate of mobile host

The message has two subsections. The first section is a signed element that includes the nonces and timestamp and the identities of the sender FA_{11} and the receiving
LMSP along with the group id G. This element is hashed and signed using the private key of FA_{11}.

The second section of the message contains the MH token that was sent as part of message (1). This token identifies the actual sender of the message. The two-signed components are linked via the identity of the MH, the group G, and the nonce N_{1}.

LMSP_{1} refers to the access list of the group to determine the fate of this request. It must then respond with an access-accept or an access-deny message. The format of the access accept message is as follows:

\[
(3a) \quad \text{LMSP}_{1} \rightarrow \text{FA}_{11}: \text{CERT-LMSP}_{1}, \text{LMSP}_{1}, N_{2}, N_{1}, T_{3}, N_{3}, \text{Graft (MH, G, L_{1})}, \text{PK-MH \{K}_{G}\}, \{\text{LMSP}_{1}, \text{MH, N}_{2}, N_{1}, K_{G}, T_{3}, N_{3}, \text{Graft (MH, G, L_{1})}\}\text{SK-LMSP}_{1}
\]

Where

- CERT-LMSP_{1}: Certificate of LMSP_{1} that is currently the group manager for the group G.
- LMSP_{1}: Identity of LMSP_{1}.
- T_{3} and N_{3} are timestamp and Nonce generated by LMSP_{1}
- Graft (MH, G, L_{1}). This message indicates that MH has been accepted into the multicast group. It is a message for FA_{11} to update its group table. L_{1} is the lifetime associated with MH’s membership.
- K_{G}: A fresh group session key computed using a secure lock technique described below and it is encrypted using the public key of MH.
- The signed section includes the identity of the LMSP_{1}, associated timestamp and nonces along with the Graft message. All these parameters are hashed and signed using the private key of LMSP_{1}

This message flow contains a timestamp and nonces of messages (1) and (2) to indicate its freshness and to protect from reply attacks (as well as to indicate that it is a response of message (1). It also includes the certificate of the sender and its identity. The signed element of LMSP is called token 2. It includes the identity of the LMSP and the timestamp and nonces of MH’s initial token in message (1) to bind the original request with the corresponding response. It also includes the Graft message that explicitly states that this mobile host is now part of group G. The encrypted section contains the group session key required by the MH to communicate securely with the group.

The Access-deny message is sent when an MH is not accepted into a particular group. This is determined by the local group policy that governs its access control list. The Access-deny message has the following format:

\[
(3b) \quad \text{LMSP}_{1} \rightarrow \text{FA}_{11}: \text{CERT-LMSP}_{1}, \text{LMSP}_{1}, N_{1}, N_{2}, T_{3}, N_{3}, \text{MSG}, \{N_{1}, N_{2}, T_{3}, N_{3}, \text{MSG}\}\text{SK-LMSP}_{1}
\]
Note that this message contains almost the same parameters as in the accept message except that it does not contain the graft message and group member identities. This indicates that the mobile host MH has not been accepted into this group. Also it contains a message MSG that indicates the reason for not granting the MH the permission to be a part of this group.

The final step involves FA_{11} conveying LMSP_1’s Message (Access-Accept message) to the concerned mobile host MH.

(4) \[ \text{FA}_{11} \rightarrow \text{MH: CERT-FA}_{11}, \text{FA}_{11}, \text{CERT-LMSP}_1, T_4, N_1, \text{PK-MH} [K_G], \{N_1, T_3, \text{FA}_{11}, G, \text{MH}\}SK-\text{FA}_{11}, \{\text{LMSP}_1, \text{MH}, N_2, N_1, K_G, T_3, N_3, \text{Graft(MH,G, L_1)}\}SK-\text{LMSP}_1 \]

This message includes token 2 generated by the LMSP is message 3(a). The message format to convey Access-Deny message will have a similar syntax except that it will not contain the Graft Message. Instead it will include a message MSG that might indicate the reason for not granting the MH the permission to be a part of this group G.

Once the authentication and group registration is accomplished, the MH can continue to receive multicast packets for this group anywhere within this segment as for each segment a single LMSP acts as a local multicast service provider. These protocol steps can be further simplified by including the id of LMSP in the FA advertisement message. In fact, upon changing its location, the MH extracts the LMSP id from the foreign agent advertisement message to determine if it needs to re-authenticate to the current LMSP if it is not the same as the previous one.

If on the other hand, the current LMSP serving the segment does not act as a group manager for the group G and is not receiving packets destined for this group, it must deploy the pull approach to retrieve this information and forward the authentication and join request across to the gmLMSP of this group. Intermediate FAs cache the Access-Accept message. This will enable the packets to be delivered to the requesting LMSP from somewhere along the delivery tree.

**Case Two: Mobile host initiating a multicast group**

A mobile host (MH) initiating a multicast group creates an access control list (ACL) and a security association (SA) for the session. It then announces this group by sending an advertisement message to its current FA. The current FA registers with it the group details along with other related information with the LMSP of this segment. Each valid recipient performs an authentication process (involving itself and its current FA and LMSP) using a process similar to that described in the previous section. The LMSP then computes the group key. This group key is computed from the public keys of the participants using the scheme described below.

Several key management schemes and protocols [Ballardie 96] exist for securely distributing keys in a network environment. In this thesis, we use the secure lock technique suggested in [Chiou+ 89]. This scheme uses the Chinese remainder theorem to generate a ‘secure lock’ to lock the deciphering group session key. The secure lock is transmitted with each encryption message. Only users in the secure group can
unlock the session key. This scheme is only efficient for small groups. The Initiator (LMSP) must store the public keys of each of the participants. From the Chinese Remainder Theorem, for \( N_1 \ldots N_n \) positive, relatively large prime integers and \( R_1 \ldots R_n \), positive integers, a set of congruous equations

\[
X = R_1 \mod N_1, \ldots, X = R_n \mod N_n
\]

has a common solution \( X \) in the range of \([1, L-1]\) where \( L = N_1 \times N_2 \times N_3 \times \ldots N_n \), where \( n \) is the number of participants in the group.

The Chinese Remainder Theorem is used to generate \( X \) where \( R_i = [Ks]P_{ki} \) where the session key \( Ks \) is encrypted using the public key \( P_{ki} \). The common lock \( X \) is a function of each of the participant’s public key. Therefore, only those participants whose public keys are included in the calculation of \( X \) can unlock it.

The dynamic addition and deletion of group members can be carried out as follows. Every time there is a change in the group membership, the gmLMSP can recreate the common \( X \) and modify the group to include or exclude certain participants form future communications. As far as the storage requirements are concerned, the gmLMSP who is the creator of the lock must store the public key of each of the participants. The decipherment of the session key \( Ks \) for each participant is fairly efficient. The scheme is a centralised one as the computation of \( X \) is restricted to a single entity thereby offering better control; however it does not scale well to large groups.

**Case Three: MH leaving a group**

When a mobile host MH leaves a particular multicast group it needs to send an explicit IGMP v2 leave message to its FA. This message includes the identity of the mobile host along with its group membership details such as the group id and the designated group manager (LMSP). Upon authenticating this request, this message is routed to the LMSP. LMSP must then recomputate the group session key eliminating the keying details of MH.

**6.4.2.5 Anonymity**

The mobility of a node raises a unique problem of anonymity that is absent in static wired networks. A mobile node must be able to seamlessly roam and attach to foreign networks. In order to access services in this foreign network the node must be able to first identify some authority in this network and be able to authenticate itself to this authority. In mobile IP architecture this authority is a FA. The FA must then inform the node’s HA of the node’s current whereabouts. This is necessary, as the home campus needs to forward the mobile host’s incoming data to the foreign domain. However, if the identity of the node MH is revealed in each visited network, a set of malicious FAs may cooperate together to monitor the movements of MH. In some situations this may not be acceptable. To prevent such an attack from happening, MH must be able to hide its true identity but it must still be able to authenticate and convince a FA of its credentials. Also the home agent must be able to identify MH’s real identity from a FA’s message in order to update the location information pertaining to this MH. We see that the location management is tightly coupled with
the problem of anonymity. In an anonymous environment the identity and the movements of MH is only disclosed to relevant entities. By withholding the identity of the MH from all entities makes it difficult to keep track of the whereabouts of MH and therefore one may not be able to establish legitimate communication session with this MH.

This leads us to the key question that governs certain design related choices with respect to node anonymity: with respect to whom must the node remain anonymous?

In the IETF Mobile IP architecture [Perkins+ 97], the home agent is responsible of the whereabouts and the movements of a mobile host. In this model, the HA is the only entity that can be trusted to maintain a binding between MH’s true identity and its ephemeral identity. The true identity of the mobile host is anonymous with respect to the foreign network and therefore the foreign agent where this node is currently located. The MH can cooperate with its HA to generate a travelling alias that MH can use in foreign networks. However, having a single alias is not useful as the attacker might use it to monitor MHs’ movements thereby making intelligent guesses about MH’s true identity. The solution would be to assign a series of aliases to MH. Some design choices are available for the generation of aliases.

Case One: The MH in cooperation with HA generates a series of aliases prior to its movement. In each visited zone it uses a unique alias. Once this list of aliases is exhausted, the MH must come back to its home zone to generate the next alias list. The drawback is that this scheme places a restriction on the seamless movement of MH. In order to avoid this problem, the MH can reuse aliases randomly from the existing list. Another problem is that the MH must always cache the alias list with itself and this may not be a secure option.

Case Two: The MH in cooperation with the HA generates an initial alias prior to its movement. At each visited zone, the MH and the HA negotiate a new alias in a secure fashion that can be used in the next visited zone. This way the MH can continue to move without being constrained in any way and also it need not cache the alias list with itself.

We implement our scheme using case two. The regional topology identifies the foreign domain in the location registration architecture. We refer to this as a zone. Prior to MH’s movement an initial alias needs to be generated. The HA generates a random number R. R is a constant. Using R and the identity of the mobile host (MH), HA computes the first alias (A1) in the following way:

\[ A_1 = h(f(MH, R)) \]

h is assumed to be a strong one-way hash function such as the Secure Hash Standard (SHA)

Both the HA and the mobile host know R. This alias is generated by the HA soon after the mobile host registers in its home zone. The HA maintains an entry for this mobile host and its alias in its directory. It also generates a token for this mobile host as defined below. This token enables the mobile host to authenticate itself to a foreign zone. The token format is as follows

Where:

- {<>}SK-Home-HA denotes that the contents of token are signed using the private key of Home-HA
- A is the alias that corresponds to a mobile host identity
- PK-A is the public key of A (public key of mobile host MH)
- Home-HA: is the identity of the home HA that generated the token.
- CERT-Home-HA is the certificate of the home HA.
- Time, validity: Validity is the time period for which the token is valid. Time denotes the time the token has been generated

The token sent serves as a pseudo certificate for the MH. It can present this token in its first visit to a foreign zone. This token proves that a mobile host A is under the jurisdiction of a HA representing a zone. Some certification authority vouches the credentials of this HA. The certificate of the HA indicates this. Validity, period allocates a lifetime for this token. It serves the same purpose as a certificate lifetime; however, the time span for this period is comparatively short. Since the token is signed using the private key for the HA, it binds this token to its originator (HA).

Let us now consider the authentication process in a foreign domain.

1. The mobile host (MH) recognises that it is in a foreign domain by means of the beacon message sent by some foreign agent known as FA. Having identified that it is in a foreign zone, MH undergoes an authentication process with this FA before registering itself in the foreign zone. This authentication process is as follows:

   (1) A -> FA, A, T, N, FA, Token, Msg, {T, N, FA, Token, Msg, A}SK-A

   The use of message field Msg is host specific. In this case, the host may use this field to convey to MSR the multicast group(s) that it belongs to.

2. FA can verify the signature using MH’s public key obtained from the token. This is necessary to prove that message actually originated from A and not from anyone masquerading as A. FA verifies the credentials of A by verifying the
signature of its HA using the HA’s public key procured through the certificate embedded in the token.

3. The registration process involves the foreign MSR informing the home MSR of the current location of MH. The foreign MSR constructs the following message for the home MSR of the mobile host.

(2) \( FA_1 \rightarrow HA: \text{CERT-FA}_1, FA_1, HA, A_1, T_1, N_1, \text{Token}, \{HA, A_1, T_1, N_1, \text{Token}\}SK-FA_1 \)

The authentication token of \( A_1 \) is signed by the foreign MSR. This is necessary for \( FA_1 \) to prove to the HA that \( A_1 \) is actually in its zone. This is because alias \( A_1 \) is being used for the first time by the mobile host MH. Only MH could have supplied this credential as itself and HA are the only entities that are aware of this identity.

4. Upon receiving this message, the HA verifies the credentials of \( FA_1 \) and the presence of MH as \( A_1 \) in its domain. It then creates a new alias \( A_2 \), which will be used by the mobile host when it moves to the next foreign zone. It completes the registration process by sending this new alias as part of a new token (Token’) to the \( FA_1 \).

(3) \( HA \rightarrow FA_1: HA, N_1, T_2, N_2, A_1, \text{[Token’]}PK-MH \{FA_1, N_1, T_2, N_2, A_1\}SK-HA \)

Upon receiving this message, the \( FA_1 \) verifies the signature of HA using the public key of HA obtained from its certificate. Note that the contents of the new token Token’ (including the new alias \( A_2 \)) is not visible to the current \( FA_1 \) as it is encrypted with the public key of the mobile host MH.

5. Finally the registration process is completed by \( FA_1 \) forwarding the new token to the mobile host

(4) \( FA_1 \rightarrow A_1 (MH): FA_1, A_1, T_3, N, \text{[Token’]}PK-MH, \{A_1, T_3, N, \text{[Token’]}PK-MH\}SK-FA_1 \)

6. The new alias \( A_2 \) in the token Token’ for use in the next signed foreign zone is generated as follows:

\[ A_2 = h(f(A_1, MH, R)). \]

The alias \( A_2 \) is a function of alias \( A_1 \) and the identity of mobile host. The forthcoming aliases would be computed as follows:

\[ A_3 = h(f(A_2, MH, R)). \]
\[ A_4 = h(f(A_3, MH, R)) \text{ and so on.} \]

Thus the mobile host acquires a new alias \( A_n \) to be used in the next zone by using its current alias \( A_{n-1} \). This is done in the registration phase to maintain a strong
binding between the mobile host and its home MSR. The same procedure is repeated when moving to the next foreign zone. This approach makes the aliases independent of the foreign domain and maintains a tight synchronisation between the mobile host and its home authority.

This scheme conceals the identity of the mobile host from the outside attackers and foreign domains. But it is not intended to conceal the identity from the home agent of the mobile host and from the members of the multicast group to which the host belongs. Such an approach is realistic and satisfies the general spirit of mobile IP architectures.

6.5 Multicast Extensions to Columbia Mobile IP

This scheme proposed by [Acharya+ 96] proposes a multicast extension scheme for the Columbia Mobile IP scheme [Ioannidis+ 91]. The crux of the problem in multicasting to/from mobile hosts in a campus is that though all MHs and wireless interfaces of MSRs within a campus share a common subnet address, link-layer connectivity amongst MHs and a MSR is present only within a single cell. Thus, for correct routing of multicast datagrams, [Acharya+ 96] provide an abstraction of link-layer connectivity amongst all MSRs and MHs within a campus. The objective is to ensure that a multicast transmission from any MH should reach all MHs regardless of their location within the campus, and wireless interfaces of all MSRs.

The concept of unicast tunnels is extended to form a multicast tunnel or MTUNNEL that links all the MSRs within a campus. The MTUNNEL provides an abstraction of link layer connectivity among the MSRs. The MTUNNEL uses a reserved multicast address for the all-MSR group and is used to forward multicast datagrams and IGMP [RFC 1112] messages from one MSR to all the other MSRs using IP within IP (version 4) encapsulation. The encapsulating IP header for a packet sent on the MTUNNEL contains the all-MSR address in its destination field and the forwarding MSR in its source field.

The MSR must determine the MH's current location within the campus before the packet can be forwarded to the appropriate cell. Instead of multiple unicasts, the MTUNNEL can be used for this purpose as well. The functionality of MTUNNEL is exploited to facilitate correct routing of multicast datagrams to/from mobile hosts.

The scheme has certain limitations. It assumes static multicast groups i.e., that the membership of any group does not change during the group’s lifetime. The scheme implies that that even if the multicast tunnels share common physical links, multiple datagrams will still be transmitted on those links. The focus here is solely on host mobility and only those multicast messages are considered that are addressed only to mobile hosts. If a host is not a member of a group, it can multicast only if it is local to a multicast router that belongs to the host view.

In this section we shall describe a secure multicast framework for the Columbia Mobile IP scheme [Ioannides+ 93].
6.5.1 Multicast Issues in Columbia Mobile IP

The basic conflict in combining IP multicast with Columbia Mobile IP is as follows: Columbia Mobile IP splits the virtual subnet across multiple cells supported by MSRs with link layer connectivity only between a MSR and MHs local to its cell [Acharya+ 95]. In contrast, IP multicast implicitly assumes that if there are multiple routers connected to a subnet, then a link layer transmission from any host on the subnet reaches all routers and hosts on that subnet. This implicit assumption does not hold true in the presence of a logical mobile subnet physically partitioned among multiple MSRs. A multicast transmission from a mobile host reaches only its local MSR and other MSRs do not receive the link layer transmission directly from the mobile host since the mobile hosts is not local to their cells. This in turn drastically affects multicast routing to/from mobile hosts. In particular, in [Acharya+ 95] present some cases on multicast routing that highlight this problem. We briefly describe these scenarios in the this section.

To understand some of the critical issues associated with carrying out multicasting in Columbia mobile IP we shall consider the following scenarios as described in [Acharya+ 95]:

- **Mobile Host (MH) as a Multicast Source.**
- **Mobile Host (MH) sending from a Foreign Campus.**
- **Mobile Host (MH) as a Multicast Recipient.**

**Scenario One: MH as a multicast source**

The problem encountered here can be best described with the help of the figure 12. Assume that there exists a multicast group having both mobile and static hosts as its members. The membership list consists of static hosts $H_1$ and $H_2$ and mobile hosts $MH_1$, $MH_2$ and $MH_3$. The arrows in the figure indicate the shortest path from the routers to the mobile subnet. When $MH_2$ under MSR$_2$ sends a multicast datagram to group $G$ packets will be forwarded to Net$_5$ and then to Net$_2$. However the packets will not be forwarded by R$_1$ to Net$_1$ since the packet did not arrive on the interface on its reverse forward path to the source. As a result the static host $H_1$ and the mobile host $MH_1$ will not receive a copy of the packet. In a nutshell, this problem occurs because for R$_1$, the interface to Net$_1$ is configured as the shortest path towards the mobile subnet. Hence all packets that arrive via Net$_2$ get discarded.

**Scenario Two: MH sending from Foreign Campus**

When a mobile host MH visits a foreign campus it can either continue to use its home address or a temporary address called nonce. If the MH uses its home address, all the packets originating at MH will be discarded by the foreign campus since they arrive at those routers on the non-shortest path to the MH’s home subnet.
Scenario Three: MH as a Multicast Recipient

A mobile host (MH) that is a member of a multicast group G when moving to a new cell may encounter delay in receiving packets due to the delay introduced by IGMP membership query procedures. Also due to a combination of the TTL value used by the source to send a datagram and threshold of the links in the multicast tree, it is possible that two mobile hosts, though part of the same subnet are at different distances from a common source. Also a mobile host can switch cells and therefore may not receive a copy of a packet since it would have exceeded the TTL value.

To solve these issues to ensure correct routing of multicast datagrams, an abstraction of link layer connectivity amongst all MSRs and MHs within a campus must be facilitated. Such a multicast framework called MTUNNEL is proposed in [Acharya+ 95]

6.5.2 MTUNNEL: Multicast Extensions for Columbia Mobile IP

The solution suggested in [Acharya+ 95] consists of ‘healing’ the partition amongst the MSRs using a predefined multicast tunnel. In other words, the concept of unicast tunnels used in mobile IP and IP multicast is extended to form a multicast tunnel or MTUNNEL that links all MSRs within a campus. The purpose of this tunnel is to provide an abstraction of link layer connectivity among the MSRs.

The multicast extension in [Acharya+ 95] uses the Distance Vector Multicast Routing Protocol (DVMRP) [RFC 1075]. DVMRP constructs a source-rooted multicast delivery tree using variants of reverse-path broadcasting algorithm (RPB). The major difference between RIP and DVMRP is that RIP is concerned with calculating next
hop to a destination, while DVMRP is concerned with computing the previous hop back to the source. In mrouted 3.0, the RPM (reverse-path multicasting) algorithm is deployed. The DVMRP forwards the packets away from a multicasting source along a group’s RPM tree. The general name of this technique is reverse-path forwarding.

This abstraction along with appropriate modification to the IGMP [RFC 1112] guarantees reliable routing of datagrams from mobile hosts to all group members; it also ensures that an MH experiences no delay in receiving datagrams regardless of the mobility within the campus. This MTUNNEL uses a reserved multicast address for all MSR group and is used to forward multicast datagrams and IGMP messages from MSR to all other MSRs using IP within IP encapsulations. The encapsulating IP header for a packet sent on the MTUNNEL contains the all-MSR address in its destination field and the forwarding MSR address in its source field.

For each of the scenarios mentioned above, [Acharya+ 95] show how MTUNNEL ensures correct routing of multicast datagrams. We shall examine then one by one.

**MH as a Multicast Source**

When a mobile host (MH) forwards a datagram, its local MSR encapsulates it and sends it via the MTUNNEL to all MSRs within the campus in addition to forwarding the original datagram to its wire-line interfaces as dictated by DVMRP. This has the same effect as the MH sending a multicast datagram on a single physical segment connecting all other mobile hosts and MSRs. This scheme will ensure that every other member of the group receives the datagram.

**MH residing in the Foreign Campus**

In this case, as mentioned above, the problem arises due to the fact that MH uses its home address. To address this problem, the datagram is tunnelled from the mobile host to any of its MSRs using standard unicast routing procedures. The home MSR at the tunnel end point processes the decapsulated multicast datagram as if it originated from its local wireless cell.

**MH as a Multicast Recipient**

The IGMP protocol is modified to generate two reports, one for the local cell, and the other being a group specific one for each MSR via MTUNNEL. This reduces the latency incurred by the mobile host when it switches its cell location as a MSR will not prune itself from a multicast tree even if it has no local members. The problem of TTL is solved by a MSR delivering a packet into the MTUNNEL and recipient MSRs only forwarding the received packet to their local cells.

The multicasting approach presented in [Acharya+ 95] is scalable and addresses issues such as routing of datagrams, correct delivery of datagram to intended recipients and delay factor when receiving datagram in a different cell. However it does not address security and privacy issues. We now extend this model and consider the provision of secure multicasting service.
6.5.3 Secure Multicast Architecture for Columbia Mobile IP

The following principles and assumptions are used in the formulation of our security model:

- We assume that a public key infrastructure is in place in the form of a certification authority (CA) or a hierarchy of certification authorities for a purpose of authentication and public key distribution. A certification authority is a trusted entity that verifies the identity of a participating entity, allocates a distinguished name to it and vouches for the identity by signing a public key certificate for that entity using a private key.

- Every host is initially registered in the campus. It receives a certificate that is signed by some CA that is local to the host.

- Each MSR has a public key and maintains a cache of the public keys of other MSRs in the campus.

- Each router in the wired network has a public key and maintains in its cache the public key of its neighbour.

- Each MSR and its next hop router on the wired network have a each other’s public keys cached within them.

- An MSR is essentially a router capable of handling mobility. It may use routing protocols such as RIP to exchange routing information with other MSRs periodically. If the MSR does not exchange such information periodically and does not answer to query requests from other MSRs, then it is considered to be non-operational.

- All MSRs in the campus share a group key that is used to encrypt reserved multicast address (for the MSR group) and all multicast datagrams and IGMP messages from one MSR to another.

- If an MSR is found to be non-operational, other MSRs need to recompute the group key. This involves eliminating the non-operational MSR.

- Each multicast group has a designated member whose responsibility is to determine who has access to belong to this group. Normally, this entity is the initiator of the group. If the initiator leaves the group, a new designated member can be chosen via an election algorithm or some other equivalent manner.

- Each MSR maintains a binding between the group identity and the identity of the designated member.

- Each MSR is trusted for authenticating a mobile user based in public key information and for maintaining the relevant multicast group information.

- The beacon message if an MSR includes its public key certificate.
• Our security model is based on IGMP v2. [RFC 2236]

We describe the security extensions by considering the following stages:

• Stage 1 considers a mobile host joining an existing multicast group or initiating
  the creation of a multicast group or leaving a multicast group.

• Stage 2 considers the movement of the mobile host to a foreign campus.

6.5.3.1 Stage 1: MH joining an existing multicast group, initiating the creation of a multicast group, leaving a multicast group.

Case 1: MH joining an existing multicast group

Consider the situation where a mobile host MH wishes to join an existing multicast
group G. The process of registering with a multicast group is as follows. The MH first
obtains the public key of the receiving MSR (MSRx) from the certificate in the
beacon message that the MSR sends. The MH then sends a join message. This join
message is typically an IGMP v2 protocol message [RFC 2236] that is extended it to
include security-related information as follows:

\[
\text{MH} \rightarrow \text{MSRx}: \text{CERT-MH, MH, } T_1, N_1, \text{MSRx, G}, \\
\{T_1, N_1, \text{MH, MSRx, G}\}_{SK-MH}.
\]

Where

• MH is the identity of the mobile host that wishes to join the multicast group.
• CERT-MH is the public key certificate of the mobile host
• MSRx is the identity of the receiving MSR
• T, N: Time stamp and Nonce generated by MH
• \{…\} SK-MH: This notation implies that the contents within \{…\} are hashed and
  signed using the private key of MH, SK-MH. In this case, it includes the nonce,
timestamp, identity of the MSR, and the address of the multicast group (G) that
MH wishes to join. The signed element is referred to as the token of MH.

The objective of message flow (1) is for the mobile host MH to join the multicast
group G. To do this, it must authenticate itself to both MSRx as well as the group
initiator. For this purpose it provides a signed element that contains its request duly
signed under its private key. The message contains a signed time stamp T and nonce
N to prove its freshness and to protect against reply attacks, the signed element also
includes the identity of the MH and the MSRx along with group id G that the mobile
host wishes to join.

Upon receiving this message, the receiving MSRx verifies the certificate and uses the
public key of the MH recovered from the certificate to verify the signed element. It
checks the integrity of the message and whether MSRx itself is the intended recipient.
It also checks the time stamp and nonce to establish whether the message is fresh.
Then MSRx refers to its multicast table entries to identify the appropriate designed
member for this group G. It then constructs the following message that is dispatched via the MTUNNEL and the wired interface of MSR_X:

(2) MSR_X -> DMH: CERT-MSR_X, MSR_X, DMH, T₁, N₁, T₂, N₂, MH, CERT-MH,
     {T₂, N₂, MH, G, MSR_X}SK-MSR_X                  SL-2
     {T₁, N₁, MH, MSR_X, G}SK-MH                      SL-1

Where

- CERT-MSR_X is the certificate of the sending MSR
- T₁, N₁: Time stamp and Nonce generated by MSR_X
- CERT-MH: Certificate of mobile host

The objective of message flow (2) is for MSR_X to authenticate itself to the multicast initiator DMH as well as to securely convey MH’s credentials and request for DMH to validate. For this purpose it includes two signed elements. The purpose of signed element SL-2 is for DMH to authenticate the sending MSR MSR_X. In addition it also includes the identities of MH and the G under the signature of MSR_X to prove that MSR_X indeed validated this request. It also includes the nonce and timestamp to thwart replay attacks.

The signed element SL-1 is from message flow (1) for DMH to validate the credentials of MH. The two-signed components are linked via the identity of the MH and the group G.

As mentioned before, by using a combination of MTUNNEL and wired interface routes, the message gets delivered to the intended designated host. The intended host responds back with an access accept or access-deny message which is delivered back to the sending mobile host.

The format of the access accept message is as follows:

(3) DMH-> MSR_X: CERT-DMH, DMH, N₁, N₂, T₃, N₃, Graft (MH, G),
     {T₂, T₁, N₂, N₃, MSR_X, DMH, K_G}SK-DMH,                  SL-3
     PK-MH[[T₁, N₁, T₃, N₃, K_G], DMH, MH]SK-DMH]               ESL-1

Where

- CERT-DMH is the certificate of designated member host of the group.
- DMH is the identity of the sender who is the designated member or the initiator of this multicast group.
- T₃ and N₃ are a timestamp and a nonce generated by DMH
• Graft (MH, G). This message indicates that MH has been accepted into the multicast group. It is a message for MSRs and other routers to update their multicast tables.

• $K_G$: A fresh group session key computed using a secure lock technique described below and it is encrypted using the public key of MH.

The objective of the message flow (3) is for the group initiator DMH to authenticate itself to MSR$_X$ as well as the MH and to convey to these entities MH’s acceptance into the group. For this purpose it includes two encrypted and two signed elements. It also includes a Graft message to explicitly convey that MH is now a member of the group. The purpose of the signed element SL-3 is for binding MSR$_X$’s message with DMH’s response. It includes timestamp and nonce from message 2 to prevent against replay attacks. The encrypted element is used for authentication and to securely convey the group key $K_G$ to MSR$_X$. The encrypted and signed element ESL-1 is for DMH to authenticate itself to MH. It includes nonce and timestamp from message (1) to bind MH’s request with DMH’s response and to prevent against replay attacks. It is signed using the private key of DMH and encrypted under the public key of MH. The encrypted element EL-2 is used for secrecy and authentication. It conveys securely the group key KG to MH. Additionally, it includes an explicit Graft message to convey MH’s inclusion into the group. This message is encrypted under the public key of MH.

The Access-Deny message is sent when an MH is not accepted into a particular group. This is similar to the message flow (3) with the exception that it does not include the group key KG and Graft message. Instead it contains a message that specifies the reason for rejecting MH’s request.

The final step involves MSR$_X$ conveying DMH’s Message (Access-Accept message) to the concerned mobile host MH.

(4) CERT-MSR$_X$, MH, T$_1$, T$_4$, N$_1$, N$_4$, MSR$_X$, MH, G, 
{\{T$_1$, T$_4$, N$_1$, N$_4$, MH, MSR$_X$, G\} SK-MH} SL-4

PK-MH[\{DMH, MSR$_X$, T$_1$, N$_1$, K$_G$, T$_3$, N$_3$\}SK-DMH] ESL-1

PK-MH[MH, G, T$_1$, T$_3$, K$_G$, Graft (MH, G)] EL-2

The objective of message flow (4) is for MSR$_X$ to authenticate itself to the MH and to securely convey DMH’s response to MH. For this reason it includes the encrypted and signed element ESL-1 and the encrypted element EL-2 from message flow 3. It also includes a signed element that contains the nonce and the timestamp from message flow (1) to bind the original request of MH with its response.

Case 2: MH initiating a multicast group

A mobile host (MH) initiating a multicast group creates an access control list (ACL) and a security association (SA) for the session. It then announces this group by sending an advertisement message across the internetwork. The announcement may be advertised to potential members by directing it to a particular multicast address reserved for receiving session announcements (SAP) or alternatively invitation
protocols such as SIP (session initiation protocol) may be used to convey the announcement to a specific group [Kruus 88]. Each valid recipient performs an authentication process (involving itself and its current MSR). The request is passed on to the initiating host, which then computes the group key. This group key is computed from the public keys of the participants using the scheme of [Chiou+ 89] described later in section 6.4.2.4.

**Case 3: MH leaving a group**

When a mobile host leaves a particular multicast group it needs to send an explicit IGMP v2 exit message to its MSR. This message includes the identity of the mobile host along with its group membership details such as group address and the designated host address. Upon authenticating this request, this message is routed to the designated host. The designated host then recomputes the common solution X, which now does not include the public key of the host that has left.

**6.5.3.2: MH moving to a foreign campus**

The following occurs in an inter-campus movement of a mobile host.

1. Having identified that it is a foreign domain, the MH undergoes an authentication process with the local MSR before registering itself in the foreign campus.

2. The registration process involves the foreign MSR informing the MSR of the home campus of the current location of the MH.

3. The MH also informs the foreign MSR of any multicast groups it currently belongs to. Alternatively, as soon as it detects a mobile host entering its cell the MSR sends a membership list that consists of groups that have members local to the cell. If the current foreign campus is already registered with this multicast group, then all the MSRs within this foreign campus get the multicast datagrams and hence they get delivered to the MH straightway. If not, there is an initial delay to the MH getting registered in this campus. All MSRs update their multicast tables with this information and there after the MGH gets datagrams delivered to it. All multicast datagrams are then tunnelled to the current location of the MH from its home campus.

The problem of anonymity in the Columbia Mobile IP scheme is similar to the one that we handled in the IETF Scheme. The intuitive solution to this problem is to assign a travelling alias to every mobile host when it is away from home campus. Then the key question is should the alias be fixed or should it be continuously changed? If the alias is fixed, then an attacker, who is closely monitoring the host’s movement may still be able to associate the alias with the true identity of the host. If the alias is constantly changing then it becomes difficult for the attacker to associate these different aliases to its true identity. This also makes it almost impossible for a set of foreign campuses to link the entire set of movements of the mobile host. However, in such situation how does a home–MSR associate each of these aliases of the mobile host to its true identity? Also the use of aliases prohibits the mobile host from using its certificate in the authentication process as certificates normally vouch for the true identity of the user. The crux of this discussion leads to the following
requirement and design principle: there should be a set of procedures or mechanisms in place by means of which a mobile host can pursue its nomadic movements by using different aliases and yet be able to authenticate itself to the foreign campus by remaining under the jurisdiction of its home campus. In this section, we present a scheme for facilitating anonymity, which aims to fulfill this requirement.

The MSR in a home campus generates a random number R. R is a constant. Using R and the identity of the mobile host (MH), MSR computes the first alias ($A_1$) in the following way:

$$A_1 = h(f(MH, R))$$

In this computation, $h$ is assumed to be a strong one-way hash function such as the Secure Hash Standard (SHA)

Both the MSR and the mobile host know R. This alias is generated by the MSR soon after the mobile host registers in its home campus. The MSR maintains an entry for this mobile host and its alias in its directory. It also generates a token for this mobile host as defined below. This token enables the mobile host to authenticate itself to a foreign campus. The token format is as follows

Token =\langle A_1, \text{PK-}A_1, \text{Home-MSR, CERT-Home-MSR, Time, Validity} \rangle, \langle\rangle_{SK-\text{Home-MSR}}

Where:

- $\langle\rangle_{SK-\text{Home-MSR}}$ denotes that the contents of token are signed using the private key of Home-MSR
- $A_1$ is the alias that corresponds to a mobile host identity
- PK-$A_1$ is the public key of $A_1$ (public key of mobile host MH)
- Home-MSR: is the identity of the home MSR that generated the token.
- CERT-Home-MSR is the certificate of the home MSR.
- Time, validity: Validity is the time period for which the token is valid. Time denotes the time the token has been generated

The token sent serves as a pseudo certificate for the mobile host. It can present this token in its first visit to a foreign campus. This token proves that a mobile host $A_1$ is under the jurisdiction of an MSR representing a campus. Some certification authority vouches the credentials of this Home-MSR. This is indicated by the certificate of Home-MSR. Validity, period allocates a lifetime for this token. It serves the same purpose as a certificate lifetime; however, the time span for this period is
comparatively short. Since the token is signed using the private key for the MSR, it binds this token to its originator (Home-MSR).

Let us now consider the authentication process in a foreign domain.

7. The mobile host (MH) recognises that it is in a foreign domain by means of the beacon message sent by some foreign MSR known as F-MSR. Having identified that it is in a foreign campus, MH undergoes an authentication process with this F-MSR before registering itself in the foreign campus. This authentication process is as follows:

   (5) $A_1 \text{ (MH)} \to F-MSR; A_1, T_1, N_1, F-MSR, Token, Msg, \{T_1, N_1, F-MSR, Token, Msg, A_1\}_{SK-A_1}$

The use of message field Msg is host specific. In this case, the host may use this field to convey to MSR the multicast group(s) that it belongs to.

8. F-MSR can verify the signature using MH’s public key obtained from the token. This is necessary to prove that message actually originated from $A_1$ and not from anyone masquerading as $A_1$. F-MSR verifies the credentials of $A_1$ by verifying the signature of its home MSR using the home MSR’s public key procured through the certificate embedded in the token.

9. The registration process involves the foreign MSR informing the home MSR of the current location of MH. The foreign MSR constructs the following message for home MSR of the mobile host.

   (6) $F-MSR \to \text{Home-MSR: CERT-F-MSR, F-MSR, Home-MSR, A_1, T_2, N}_2, \text{Token,}$
   $\{\text{Home-MSR, A}_1, T_2, N_2, \text{Token}\}_{SK-F-MSR}$

Authentication token of $A_1$ is signed by the foreign MSR. This is necessary for foreign MSR to prove to Home MSR that $A_1$ is actually in its campus. This is because alias $A_1$ is being used for the first time by the mobile host MH. Only MH could have supplied this credential since this MH and the Home-MSR are the only entities aware of this identity.

10. Upon receiving this message, the home MSR verifies the credentials of foreign MSR and the presence of MH as $A_1$ in its domain. It then creates a new alias $A_2$, which will be used by the mobile host when it moves to the next foreign campus. It completes the registration process by sending this new alias as part of a new token (Token’) to the F-MSR.

   (7) $\text{Home-MSR} \to \text{F-MSR: Home-MSR, N}_2 T_2, N_3, T_3, A_1,$
   $\text{PK-MH[Token’]},$
   $\{\text{F-MSR, N}_2, T_2, N_3, T_3\}_{SK-Home-MSR}$

Upon receiving this message, the foreign MSR verifies the signature of home MSR using the public key of home MSR obtained from its certificate. Note that
the contents of the new token Token’ (including the new alias A2) is not visible to the current F-MSR as it is encrypted with the public key of the mobile host MH.

11. Finally the registration process is completed by F-MSR forwarding the new token to the mobile host

(8) F-MSR -> A1 (MH): F-MSR, A1, N1, N4, T1, T4, PK-MH [Token’],
PK-MH [{A1, N1, N4, T1, T4, (Token’)}SK-F-MSR]

12. The new alias A2 in the token Token’ for use in the next signed foreign campus is generated as follows:

\[ A2 = h(A1 \oplus MH \oplus R) . \]

The alias A2 is a function of alias A1 and the identity of mobile host. The forthcoming aliases would be computed as follows:

\[ A3 = h(A2 \oplus MH \oplus R). \]
\[ A4 = h(A3 \oplus MH \oplus R) \text{ and so on.} \]

Thus the mobile host acquires a new alias A_N to be used in the next campus by using its current alias A_{N-1}. This is done in the registration phase to maintain a strong binding between the mobile host and its home MSR. The same procedure is repeated when moving to the next foreign campus. This approach makes the aliases independent of the foreign domain and maintains a tight synchronisation between the mobile host and its home authority.

Just like the scheme for IETF Mobile IP architecture [Perkins 98], this scheme also aims to conceal the identity of the mobile host from the outside attackers and foreign domains but not from home MSRs.

Consider the following two cases:

- In the first case the mobile host registers in the multicast group G when it is in its home domain. The identity of the mobile host is disclosed at the time of registration. When the host moves to a foreign domain it acquires an alias A1. It uses this alias to access the services offered by G. Even though it has acquired this new alias it can still access the services of G as the multicast group key remains the same as the host’s key remains the same. Hence there is a possibility that the DMH and other members of the group may be able to identify the mobile host by mapping the aliases of the mobile host to its public key. However we envisage an environment where the public key to user name mapping is tightly controlled. In other words, if an entity requests the user name or the certificate of a user to the concerned authority his request is declined straightway.

- In the second case, the mobile host joins a multicast group when it is in a foreign domain. In this case, the true identity of the host is not disclosed as it is under an alias say A1. Here the members of the group may be able to map
the various aliases to $A_1$ but cannot map them to the true identity of the mobile host.

### 6.6 Summary

Multicast services and mobile networks are among the emerging technologies of the last decade. In this chapter, we first examined various multicast routing paradigms proposed for mobile networks. Integration of multicast services over mobile infrastructure presents several challenges. We have described various schemes that enable multicasting in mobile-networked environment. Several research proposals have surfaced that examine secure multicast in mobile networks. In this chapter we first investigated issues of designing secure multicast services in mobile networks and thereafter described the various schemes that aim to achieve the same. In particular, we considered provision of secure multicast service in IETF mobile IP and Columbia Mobile IP schemes. The frameworks proposed in [Omar+ 00] and [Acharya+ 95] seem to provide a good basis for considering multicasting in these two types of mobile IP networks. In this chapter, we have extended these frameworks by developing security models that can be used to provide a secure multicasting service. In particular, we have considered the various phases of a mobile host joining, initiating and leaving a multicast group and have proposed appropriate security protocols. The chapter also considered secure group key generation and distribution. Finally, the chapter discussed the movement of mobile hosts between campuses and describes an alias-based authentication scheme in such an inter-domain environment.
Part D

Ad-Hoc Networks
Chapter 7

Introduction to Ad-hoc Networks

7.1 Introduction

Mobile Ad-hoc networks (MANET) are autonomous networks consisting of routing nodes (or some routing nodes with other nodes that do not route) that are free to move randomly and organize themselves arbitrarily; thus the network’s wireless topology may change rapidly and unpredictably. In other words, a MANET may simply be defined as a collection of mobile hosts that maintain interconnection without the intervention of centralised access point. Every node in the network functions as an end application node as well as a router and forwards packets on behalf of other nodes. In order to achieve this, each node must participate in an ad-hoc routing protocol that allows it to discover multi-hop paths through the network to any other node. Such a network may operate in a standalone fashion, or be connected to a larger Internet.

Two important properties of an ad-hoc network are that it is self-organizing and adaptive. ‘Self organizing’ implies that a network can be formed on the fly and then change its topology without the presence of system administration entities. The term ‘adaptive’ simply implies that an ad-hoc network can take different forms and has highly variable mobile characteristics such as power and transmission conditions, traffic distribution variations, and load balancing. In order to bring forth such an infrastructure, the participating devices must be capable of detecting the presence of other devices and perform necessary handshaking to allow communications and sharing of information and services.

Ad-hoc wireless networks offer unique benefits and versatility for certain environments and certain applications [Chakrabarti+ 01]. Since there is no fixed infrastructure, such networks can be deployed anywhere and at any time on the fly. Since these networks are not governed by the limitations imposed by a wired topology they tend to be inherently more fault resilient. All nodes are mobile and there is no fixed management entity involved in the addition and deletion of nodes from the network. Due to these inherent advantages, this networked technology generated a lot of interest in the research community and it was perceived that several agencies such as military, police, and rescue could gain from its usage. In recent years, home or mall office networking and collaborative computing with laptop computers in a small area have also emerged as other major potential areas of where the use of MANET could be useful. However there are numerous challenges such as effective routing, security, power management, mobility management, and QoS related issues that must be overcome to realise the practical benefits of ad-hoc networking. In this introductory chapter on ad-hoc networks we discuss some of these issues along with some general and foundation topics that provide context for the rest of the chapters that follow in this thesis. This chapter begins by first looking into the origins of ad-hoc networks in section 7.2. In section 7.3, the general characteristics of ad-hoc networks are examined and the challenges the designers face when designing and deploying such networks are highlighted. One specific challenge is the absence of a routing infrastructure, which is critical for a network to function. Therefore, section 7.4 provides a survey of various principle routing solutions proposed for ad-hoc networks.
In section 7.5, some applications for ad-hoc network are highlighted. Finally in section 7.6, we furnish concluding remarks.

7.2 Origins of Ad-hoc Networks

The history of ad-hoc networks can be traced back to 1972 and the DoD-sponsored Packet Radio Network (PRNET), which evolved into the Survivable Adaptive Radio Networks (SURAN) program in the early 1980s [Freebersyser+ 01]. Although ARPANet first introduced packet switching technology in the 1960s, it was not until the growth of the Internet infrastructure and the microcomputer revolution that packet radio network ideas became truly applicable and feasible. One of the original motivations for MANET is found in the military need for battlefield survivability. Soldiers must be able to move freely without any of the restrictions imposed by wired communication devices. An additional motivation for MANET is that the military cannot rely on access to a fixed, pre- placed communications infrastructure in battlefield environments. Therefore, the goal of the DoD sponsored programs was to provide packet-switched networking to mobile battlefield elements in an infrastructureless, hostile environment (such as soldiers, tanks, aircraft, etc.) forming the nodes in the network. The packet radio was the first implementation of an infrastructureless network, where nodes are mobile, including the mobile device.

The PRNET used a combination of ALOHA and CSMA approaches for medium access, and a form of distance-vector routing. SURAN significantly improved upon the radios (making them smaller, cheaper, power-thrifty), scalability of algorithms, and resilience to electronic attacks. The routing protocols were based on the principle of hierarchical link-state and were highly scalable.

In the early 1990s new developments came into existence heralding a new phase in ad-hoc networking. Use of Laptop computers proliferated along with open-source software, and viable communications equipment based on RF and infrared. Soon the IEEE 802.11 subcommittee adopted the term "ad-hoc networks." The concept of commercial (non-military) ad-hoc networking had arrived. Other novel non-military possibilities were explored and interest grew.

The DoD reinitiated the research in infrastructureless networks and its funded programs resulted in the Global Mobile Information Systems (GloMo) and Near-term Digital Radio (NTDR) schemes. The goal of GloMo was to provide office-environment Ethernet-type multimedia connectivity anytime, anywhere, in handheld devices. Channel access approaches were now in the CSMA/CA and TDMA modes, and several novel routing and topology control schemes were developed. The NTDR used clustering techniques along with link-state routing. NTDR clusters are a single level of clusters, each composed of nodes within one hop of a control node known as the clusterhead. Now used by the US Army, NTDR is the only "real" (non-prototypical) ad-hoc network in use today.

Spurred by the growing interest in ad-hoc networking, a number of standards activities and commercial standards evolved in the mid to late '90s. Within the IETF, the Mobile Ad-hoc Networking (MANET) working group was born, and sought to standardize routing protocols for ad-hoc networks. The development of routing within the MANET working group and the larger community forked into reactive (routes on-
demand) and proactive (routes ready-to-use) routing protocols [Royer+99]. The 
802.11 sub-committee standardized a medium access protocol that was based on 
collision avoidance and tolerated hidden terminals, making it usable, if not optimal, 
for building mobile ad-hoc network prototypes using notebooks and 802.11 PCMCIA 
cards. HIPERLAN and Bluetooth were some other standards that addressed ad-hoc 
networking requirements.

7.3 Ad-hoc Networks: Characteristics and Design Challenges

The following characteristics adequately depict an ad-hoc networking environment 
[Rafique 02]:

1. **Autonomous Terminal**: In MANET, each mobile node can either run in an 
autonomous mode or it can act as both a host and a router. Therefore it is not 
possible to clearly distinguish between an endpoint mobile node and routers.

2. **Distributed Operation**: There is no fixed centralised authority to manage and 
control ad-hoc networks. Therefore these functions are distributed amongst 
individual nodes in the network. This implies that a high degree of 
collaboration is necessary between the participating nodes to realise such a 
network.

3. **Routing**: MANET has provisions for single hop as well as multihop routing 
depending upon different link layer attributes and routing protocols. Single 
Hop routing is simpler with the cost of lesser functionality and applicability. 
Packets are only delivered to directly connected nodes (nodes within the range 
of each other). In multihop routing, when delivering packets from a source to 
a destination out of the direct wireless transmission range, the packets should 
be forwarded via one or more intermediate nodes.

4. **Dynamic Network Topology**: In MANET, nodes can be extremely mobile as 
a result of which the topology of the network changes frequently and 
dynamically. Such movements are unpredictable and may result in frequent 
disconnections. The challenge is to adapt to traffic and propagation conditions 
as well as the mobility patterns of the mobile network nodes. Each node upon 
changing its location must be able to collaborate with nodes at the new 
location, determine the topology, and establish routing procedures to carry out 
communication. There may be a scenario where a participating node may wish 
to have access to services outside a MANET: other MANETS or a public 
fixed network or the Internet.

5. **Unpredictable Link Capacity**: The presence of high bit error rates of 
wireless connection coupled with unpredictable topology changes subject a 
communication channel to noise, fading, and interference. Even worse, the 
path between any pair of endpoint nodes may traverse multiple wireless links 
and the link themselves can be heterogeneous.

6. **Lightweight Terminals**: In most cases, the MANET nodes are mobile devices 
with less CPU processing capacity, small memory size, and low power 
storage.

Ability to self configure coupled with a highly adaptive nature are inherent 
advantages of a MANET architecture. However, the mobility of devices, the 
limitations of the wireless medium and the ad-hoc nature of networks provide 
challenges that must be addressed. These challenges are as follows:
1. **Network Routing Problems:** In ad-hoc networks, the topology may vary at a rapid rate. The routing protocols must be able to adapt quickly to these changes and come to an equilibrium state. The protocols therefore must be reactive rather than being proactive. Existing routing protocols are unable to catch up with frequent changes in topology resulting in poor route convergence and very low communication throughput. Several new proposals for MANET have surfaced in the recent years such as [AODV], [DSR]. In principle, these protocols can be classified into two schools of thought: one that places an emphasis on route caching (table driven), and the other that places an emphasis on dynamic route discovery (on-demand).

2. **Power Consumption and Energy Conservation:** Since most traditional routing and communication protocols were designed for static hosts in wired networks they do not consider power consumption or energy conservation as an issue at all. For most of the lightweight terminals used in MANET, the communication and processing-related functions should be optimised for low power consumption. Battery lifetime is limited and the problem is further aggravated, as each participating node in the network is both a router and a communicating end node. Solutions that consider conservation of power and carry out power-aware routing will be favoured.

3. **Issues Relating to Multicasting:** Multicast conferring is a rapidly growing area in the Internet. Audio, video, and other media such as shared whiteboard data can be efficiently distributed between groups of conference participants using multicast algorithms that minimise the amount of traffic sent over the network. Traditional multicast routing protocols either use broadcast and prune approach to build the multicast tree rooted at the source or rely on core nodes where the multicast tree originates. All such methods were designed for wired environments wherein it can be assumed that routers are not mobile. This assumption does not hold true for ad-hoc networks.

4. **QoS Support:** Quality of Service in networking is concerned with providing services to the application through negotiation with the underlying network to provide a certain level of quality from the network such as throughput, bit error rates, latency, cost etc. QoS is meaningful only for a flow of packets between the source and the destination, and thus depends on the notion of a logical association or logical connection between them for the duration of flow. Also, to attain and preserve the service attributed for such a logical connection, the network must guarantee the availability of a set of resources associated with the flow. Consequently, the routers must remain aware of the logical connection and state of the flow to ensure that adequate network resources such as link bandwidth, nodal buffers, and processing power are available for the duration of the logical connection and their underlying routes. In other words, QoS guarantees can only be assured by explicit resource reservation schemes [Chakrabarti+ 01]. Providing QoS support in MANET where the network topology is constantly changing is a challenge. The inherent stochastic feature of communications quality in a MANET makes it difficult to offer fixed guarantees on the services offered to a device. Also the inability to predict the network behaviour and capabilities further aggravates this problem. An adaptive QoS model must be implemented over traditional resource reservation to support multimedia services [Rafique 02].

5. **TCP Performance:** The Transmission Control Protocol (TCP) is one of the most popular and widely used end-to-end protocols for the Internet. TCP is a
connection-oriented protocol that provides flow and congestion control functions in addition to reliable transmission feature on an end-to-end basis. The fundamental assumption that governs the principle of TCP operation is that nodes in the route are static and therefore it makes sense to perform flow and congestion control functions of the source and the destination nodes. Therefore, TCP is unable to distinguish between the presence of mobility and network congestion. TCP performance is drastically affected as underlying mobility introduces packet loss and long round trip time (RTT). Therefore, some changes and extensions are needed to ensure that TCP understands the nature of mobility and performs efficiently without affecting end-to-end communication.

6. **Service Location, Provision and Access:** The placement, provision and availability of services are critical for the proper operation of ad-hoc networks. Traditional client-server mechanisms are in place in wired networks to facilitate such services. It is challenging to achieve the same in this environment due to the presence of heterogeneous devices and machines and not every machine is capable of acting as a server. Hence the concept of a client initiating task requests to a server for execution and awaiting results may not be attractive due to limitations in bandwidth and power. Also there are issues relating to how a mobile device can access a remote service or how does the availability of a particular service get advertised in the network that need to be seriously considered. [Toh 02].

7. **Connection to external Networks:** As mentioned before, a node participating in a MANET might require access to services outside of the MANET. Some mechanisms must exist to facilitate such a connection. Should this interface be made available via a fixed gateway or is there a possibility of achieving the same using a mobile router? A mobile router can be a router that is responsible for the mobility of one or more networks moving together, perhaps on an airplane or a ship [Perkins 98]. This is a scenario where the mobile router is mobile with respect to a network, which itself is also mobile. The mobile router can connect to the wired network on behalf of a mobile network.

8. **Problems Related to Physical Layer:** The physical channel is susceptible to noise and interference and this in turn leads to relatively low bandwidth and data rates and high error rates. There are also regulations regarding the use of the radio spectrum that are currently under the control of the FCC. Most experimental ad-hoc networks are based on the ISM band. To avoid interference, ad-hoc networks must operate over some form of allowed or specified spectrum range [Toh 02].

9. **Data Link Layer Problems:** Several mobile nodes must share the same media and therefore a media access protocol is required to handle contention-based issues. Traditional MAC protocols do not take mobility into consideration and hence are unable to handle problems that are peculiar only in mobile environments such as hidden terminal problem. Two nodes are said to be hidden from one another (out of signal range) when both attempt to send information to the same receiving node, resulting in a collision of data at the receiver node [Toh 02]. They also mostly rely on a centralised coordinator, an assumption that does not hold true for ad-hoc networks. Hence they fail to scale up to meet the demands and requirements of ad-hoc networks.
Another core data link layer issue is whether packets are transmitted. Normalizing signal strength at which received signals are considered to be validly transmitted messages is the key.

10. **Security and Reliability**: The security threats that prevail in a traditional wired and wireless network environment also hold true for wireless ad-hoc networks but are further aggravated due to unpredictable and dynamic nature of such networks. There are also some security issues that are peculiar only to MANET such as a nasty neighbour relaying packets or a malicious node disrupting the routing infrastructure. Further, wireless link characteristics also introduce reliability problems due to limited wireless transmission range, the broadcast nature of the wireless medium, mobility induced packet losses, and data transmission errors.

### 7.4 Routing Protocols for Ad-hoc Networks: A Survey

#### 7.4.1 Introduction

There are several scalable and efficient routing schemes available in the Internet for forwarding packets from a source to a particular destination. However for these protocols to run over ad-hoc networks they must deal with limitations inherent in these networks such as rapid changing topology, lack of a fixed static infrastructure, high power consumption, low bandwidth, and high error rates. In the recent years several attempts have been made to adapt existing routing protocols for use in ad-hoc networks. Such MANET routing protocols are classified into two categories: proactive or table-driven and reactive or on-demand routing based on when and how the routes are discovered. In table driven routing protocols consistent and up-to-date routing information to all nodes is maintained at each node whereas in on-demand routing the routes are created only when desired by the source host. On the other hand, reactive or on-demand routing protocols have been designed to acquire routing information only when needed.

Any routing proposal for MANETs must meet two conflicting goals, that is, frequent topology updates are required to optimise routes, and while frequent topology updates result in higher message overhead and bandwidth wastage. Different routing proposals use different metrics to address these goals. Some protocols use shortest hop routing while some attempt to minimise overheads during route decision making using link quality information while some other rely on associativity based approach. Any design proposal for routing in MANETs must identify and place an emphasis on such metrics to establish an optimised trade off between the two goals.

#### 7.4.2 Table Driven Routing Protocols

In Table-driven routing protocols each node maintains one or more tables containing routing information to every other node in the network. All nodes update these tables so as to maintain a consistent and up-to-date view of the network. When the network topology changes the nodes propagate update messages throughout the network in order to maintain consistent and up-to-date routing information about the whole network. These routing protocols differ in the method by which the topology change information is distributed across the network and the number of necessary routing-related tables.
Proactive or table-driven routing protocols may again be subdivided depending on the manner in which route tables are constructed, maintained, and updated. The two primary classes are as follows [Perkins+ 96]:

- **Link State**: In this protocol, each node maintains a view of the entire network topology with a cost for each link. To keep these views consistent, each node periodically broadcasts the link costs of its outgoing links to all other nodes using a protocol such as flooding. As a node receives the information, it updates its view of the network topology and applies a minimum-cost algorithm to choose its next hop for each destination.

- **Distance Vector**: In distance vector routing, each node maintains a routing table consisting of a destination IP address, distance to the destination (number of hops) and the next node in the path. Each router periodically broadcasts this table information to each of its neighbouring routers, and uses similar routing updates received from its neighbours to update its own tables.

Some important table-driven protocols that we discuss in this section are:

- Destination Sequenced Distance-Vector Protocol (DSDV) [Perkins+94].
- Wireless Routing Protocol (WRP) [Murthy+96].
- Global State Routing (GSR) [Chen+98].
- Fisheye State Routing (FSR) [Iwata+99].
- Hierarchical State Routing (HSR) [Pei+98].
- Zone Based Hierarchical LSR Protocol (ZHLS) [Ng+99].
- Cluster Switch Gateway Routing (CGSR) [Chiang+ 97].

### 7.4.2.1 Destination Sequenced Distance-Vector (DSDV) Protocol

The Destination-Sequenced Distance-Vector (DSDV) Routing Algorithm [Perkins+94] is based on the idea of the classical Bellman-Ford Routing Algorithm with certain improvements. In DSDV, every mobile station holds a routing table listing the following:

- All available destinations,
- Number of hops to reach the destination
- Sequence number assigned by the destination node.

The DSDV algorithm differs from other distance vector algorithms by using a sequence number to distinguish stale routes from new ones and thus avoid the formation of loops. If more than one path has the same sequence number, the path with the smallest distance metric is chosen. The transmission of routing table can be time driven or event driven. In the case of time-driven update, the stations periodically transmit their routing tables to their immediate neighbours. In the case of event-driven update, a station also transmits its routing table if a significant change has occurred in its table from the last update sent.

There are two ways to send the update itself: the Full Dump approach or the Incremental Update. A full dump sends the full routing table to the neighbours and
could span many packets whereas in an incremental update only those entries from the routing table are sent that have a metric change since the last update and they must fit in a packet. If there is space in the incremental update packet then those entries may be included whose sequence numbers have changed. Choosing an option depends upon the existing network conditions. When the network is relatively stable, incremental updates are sent to avoid extra traffic and full dump are relatively infrequent. In a rapidly changing network, incremental packets can grow big so full dumps will be more frequent.

Each route update packet, in addition to the routing table information, also contains a unique sequence number assigned by the transmitter. The route labelled with the highest (i.e. most recent) sequence number is used. If two routes have the same sequence number then the route with the best metric (i.e. shortest route) is used. Based on the past history, the stations estimate the settling time of routes. The stations delay the transmission of a routing update by settling time so as to eliminate those updates that would occur if a better route were found very soon.

DSDV has an inherent limitation as it has a requirement of periodic update transmissions, regardless of the number of changes in the network topology. This effects scalability of the protocol by limiting the number of nodes that can connect to the network since the overhead grows as O(n^2).

7.4.2.2 Wireless Routing Protocol (WRP)

The Wireless Routing Protocol (WRP) [Murthy+96] is a table-based distance-vector routing protocol. Each node in the network maintains the following:

- **Distance Table**: Contains the distance of each destination node via each neighbour of the source. It also contains the information about the downstream neighbour of the source’s neighbour through which this path is realized.

- **Routing Table**: Contains the distance of each destination node from the source node, the predecessor and the successor of the node on this path. It also contains a tag to identify if the entry is a simple path, a loop or invalid. Storing the predecessor and successor in the table is beneficial for detecting loops and avoiding counting-to-infinity problems.

- **Link-Cost Table**: Contains the cost of the link to each neighbour of the node and the number of timeouts since an error-free message was received from that neighbour.

- **Message Retransmission List**: The Message Retransmission list (MRL) contains information to let a node know which of its neighbour has not acknowledged its update message and to retransmit an update message to that neighbour.

Update messages are either sent periodically or when the link status changes. The receiving nodes must acknowledge this update message. They also use this information to identify better paths towards various destinations. The new paths computed are sent back to the original nodes for them to update their tables. The updating process also occurs when a path in an update message is better than a path in the existing routing table. Upon receiving an acknowledgement from the receiving node, the sending node must update its MRL. A unique feature of this algorithm is
that it checks the consistency of all its neighbours every time it detects a change in the links of any of its neighbours. Such a scheme helps eliminate looping situations in an efficient way and also has a fast convergence property.

WRP has lower time complexity than DSDV since it only informs neighbouring nodes about link status changes. However in terms of communication complexity, since DSDV and WRP use distance vector shortest path routing as the underlying routing protocol, they both have the same degree of complexity during link failures and additions [Royer+ 99].

The WRP scheme has some limitations. It requires each node to maintain four routing tables, which can lead to substantial memory requirements especially when the number of nodes in the network is large [Royer+ 99]. This protocol also generates ‘Hello’ packets whenever there are no recent packet transmissions from a given node. These packets tend to consume bandwidth.

7.4.2.3 Global State Routing (GSR)

GSR [Chen+98] improves DSDV by avoiding flooding of routing messages. Each node maintains the following:

- Neighbour List: Contains the list of its neighbours
- Topology Table: Contains the link state information as reported by the destination and the timestamp of the information.
- Next Hop Table: Contains the next hop to which the packets for a destination must be forwarded.
- Distance table: contains the shortest distance to each destination node.

A change in link results in the generation of routing messages and each receiver upon receiving such a message updates its topology table if and only if the sequence number of the message is newer than the sequence number stored in the table. Having done this, the node then reconstructs its routing table and broadcasts the information to its neighbours.

The problem with this approach is that large size update messages are exchanged among neighbouring nodes periodically. Clearly this consumes a considerable amount of bandwidth when the network size becomes large.

7.4.2.4 Fisheye State Routing (FSR)

A further improvement on GSR can be seen in an approach called Fisheye State Routing (FSR) [Iwata+99]. FSR is an implicit hierarchical routing protocol. It uses the “fisheye” technique developed by [Kleinrock+ 71], where the technique was used to reduce the size of information required to represent graphical data. The eye of a fish captures with high detail the pixels near the focal point. The detail decreases as the distance from the focal point increases. In routing, the fisheye approach translates to maintaining accurate distance and path quality information about the intermediate neighbourhood of a node, with progressively less detail as the distance increases.
The FSR protocol eliminates large size update messages of GSR by frequently sending short messages that include only those nodes in close proximity and not which are farther away. As a result of this mechanism, a node gets an accurate picture of its surroundings and this picture gets less accurate as the distance from node increases. A scope defines the area for which accurate information is maintained. Even though a node does not have accurate information about distant nodes, the packets are routed correctly because the route information becomes more and more accurate as the packet moves closer to the destination.

The FSR protocol scales well to large networks as the overhead is controlled in this scheme. The FSR periodic table exchange resembles the vector exchange in Distributed Bellman-Ford (DBF) of DSDV where the distances are updated according to the timestamp or sequence number assigned by the node originating the update. However, Unlike in DSDV, in FSR link states rather than distance vectors are propagated.

One clear disadvantage of FSR is that although it produces timely updates from near stations, it creates large latencies from stations afar. However, it can be argued that the imprecise knowledge of the best path to a distant destination is compensated by the fact that the route becomes progressively more accurate as the packets get closer to the destination.

### 7.4.2.5 Hierarchical State Routing (HSR)

In [Pei+98] a soft state wireless hierarchical routing protocol called Hierarchical State Routing (HSR) is introduced. In HSR, just as in CGSR, the topology is logically partitioned into clusters and within each such cluster a node is elected to be a clusterhead. The nodes within a cluster broadcast their link information to each other. The clusterhead in turn, summarizes this information to feed it to other neighbouring clusters. This information is sent via a gateway to the neighbouring clusterhead. Unlike CGSR, each clusterhead is a member of a cluster in the next higher level. The flooding occurs from a higher to a lower level. As a result of this process, the hierarchical topology information is maintained at the lower level. Also unlike CGSR, this scheme also prescribes solution to partition the nodes into logical subnetworks under the jurisdiction of a location management server (LMS). While clustering is based on geographical relationships between the nodes, logical partitioning is based on logical and functional affinities between nodes such as tanks in the battlefield, or travelling salesmen of the same company. Each node has a hierarchical as well as a logical address. The search operation does a mapping between logical and Hierarchical addresses. Logical partitions play a key role in mobility management. The proposed mobility management scheme tracks mobile nodes, while keeping the control messages low.

In order to define logical partitions, in addition to MAC addresses, nodes are assigned logical addresses of the type <subnet, host>. These addresses have a format similar to IP, and can be in fact viewed as private IP addresses for the wireless network. Each IP subnet corresponds to a particular user group with common characteristics such as a search team in a search and rescue operation. A different mobility pattern can be defined independently for each subnet. This scheme also introduces a home agent for each IP subnet to manage the hierarchical address changes of its subnet members.
In terms of throughput, the performance of HSR is better than DSDV [Pei+ 98]. DSDV’s poor performance can be attributed to excessive channel usage by route control messages. Also, as mobility speed increases, more event-triggered updates are generated in DSDV. In terms of control overheads, DSDV propagates full routing tables whereas HSR uses much smaller tables. The authors through various simulation results show that at 100 nodes a flat routing scheme such as DSDV is untenable if the network is mobile and therefore requires rapid refresh.

HSR performance degrades as the group size increases. HSR has the highest performance when the logical subnets are identical to the group in the group mobility model. HSR has the worst performance in the case of group member size equal to 1 that is, each individual node has its own mobility pattern.

### 7.4.2.6 Zone Based Hierarchical LSR Protocol

In [Ng+99], a peer-to-peer hierarchical routing approach is proposed. In this scheme, the network is divided into non-overlapping zones. Aggregating nodes into zones conceals the detail of the network topology. Initially, each node knows its own position and therefore zone ID through the Global Positioning System (GPS). After the network is established, each node knows the low level (node level) topology about node connectivity within its zone and the high level (Zone level) topology about the zone connectivity of the whole network. A packet is forwarded by specifying the hierarchical address-zone id and node id of a destination node in the packet header. Unlike CSGR and other hierarchical protocols, there are no clusterheads in this protocol. The high level topological information is distributed to all nodes in a peer-to-peer manner. This peer-to-peer characteristic avoids traffic bottleneck, prevents single point of failure and simplifies mobility management. Location search is performed by unicasting one location request to each zone. Routing is done by specifying the zone id and the node id of the destination, instead of specifying an ordered list of all nodes between the source and the destination. Intermediate link breakage may not cause any subsequent location search. Since the network consists of non-overlapping zones in ZHLS, frequency reuse can be readily deployed.

The advantage of ZHLS is that it uses broadcasting on zone level topology, which saves more bandwidth than flooding as in other flat protocols such as AODV and DSR. Also, the chance of a virtual link breaking in ZHLS is smaller than that of a physical link breakage in a flat protocol such as AODV. So, the zone level topology in ZHLS is relatively stable. It also generates a smaller amount of communication overhead than flooding in flat protocols. Also the actual routing path is adaptable to changing topology, and a subsequent search is not required as long as the destination does not hand off to another zone. However, the disadvantage is that the hierarchical routing in ZHLS leads to suboptimal path between two nodes, and so the length of ZHLS path may be higher when compared to other flat routing protocols.

### 7.4.2.7 Cluster Switch Gateway Routing (CGSR)

Cluster Gateway Switch Routing (CGSR) [Chiang+ 97] is a cluster based routing protocol where the nodes are grouped into clusters with each cluster being governed by a clusterhead. The clustering technique provides a framework for code separation, channel access, routing and bandwidth allocation among clusters [Toh 02]. Any
distributed election algorithm can be applied to select the clusterhead. Although using a clusterhead provides some form of control and coordination, it does impose reliance from other nodes within the cluster. Clusterheads themselves can move like any other node in which case a new clusterhead must be selected. The problem with this approach is that if there is a frequent change in clusterhead networks will use up a lot of time converging to the new equilibrium state. To avoid this overhead, a least cluster change (LCC) algorithm is introduced. Using the LCC algorithm, clusterheads only change when two clusterheads come into contact, or when a node moves out of all other clusterheads.

CGSR uses DSDV as the underlying routing scheme, and hence has much of the same overhead as DSDV. However, it modifies DSDV by using a hierarchical clusterhead-to-gateway routing approach to route traffic from source to destination. Gateway nodes are nodes that are within the communication range of two or more clusterheads. A packet from the sender is first routed to its clusterhead from where it gets routed to the gateway and then on to the next clusterhead. This process goes on until the clusterhead of the destination node is reached.

One advantage of CGSR is that several heuristic methods can be employed to improve the protocol’s performance. These methods include token scheduling, gateway code scheduling, and path reservation [Chiang+ 97]. However, this scheme has some drawbacks. In CGSR the routing performance is dependent on the status of specific nodes such as the clusterhead and the gateway. Therefore the time complexity of a link failure associated with a clusterhead is higher than DSDV, given the additional time needed to perform clusterhead reselection. This scheme also has high storage requirements as in addition to the cluster member table; each node must also maintain a routing table, which is used to determine the next hop in order to reach the destination. Additional load on bandwidth is introduced as cluster member tables are periodically broadcast by each node using the DSDV protocol.

### 7.4.3 On-Demand Routing Protocols

The approach in ‘on demand routing’ protocols, unlike table-driven schemes, is to create routes as and when required. This implies that all up-to-date routes are not maintained at every node. When a source wants to send to a destination, it invokes the route discovery mechanisms to find the path to the destination. The route remains valid till the destination is reachable or until the route is no longer needed. We now discuss a few on-demand routing protocols. Some important on demand routing protocols that we examine in this section are as follows:

- **Cluster Based Routing Protocol (CBRP)** [Jiang+99].
- **Ad-hoc On-demand Distance Vector Routing Protocol (AODV)** [Perkins+99].
- **Dynamic Source Routing Protocol (DSR)** [Johnson+99].
- **Temporally Ordered Routing Algorithm (TORA)** [Park+97].
- **Associativity Based Routing (ABR)** [Toh96, Toh99].
- **Signal Stability Based Adaptive Routing (SSR)** [Dube+ 97].
7.4.3.1 Cluster Based Routing Protocol (CBRP)

In [Jiang+99], a cluster based routing protocol (CBRP) is proposed that partitions the nodes into logical clusters; each cluster governed by a clusterhead. The clusterhead schedules transmissions and allocates resources within the clusters. The selection process of a clusterhead can be centralised or distributed. In a centralised approach, the node with the lowest or highest numbered identifier or with largest number of neighbours is chosen as a clusterhead. With the distributed version, a node elects itself as a clusterhead if it has the lowest or highest numbered identifier in its neighbourhood or if it is highly connected to all of its uncovered neighbours. A clusterhead keeps information about the members of its cluster and also maintains a cluster adjacency table that contains information about the neighbouring clusters. For each neighbour cluster, the table has an entry that contains the gateway through which the cluster can be reached and the clusterhead of the cluster. Each node in the cluster is initially in an undecided state. It initially sends a “hello message” for which it must receive an acknowledgement from the clusterhead. Upon receiving such an acknowledgement it changes its state to become a member. If the undecided node times out, then it may get elected as the clusterhead if it fulfils the criteria described above. Clusterheads are changed as infrequently as possible. CBRP adopts the cluster formation algorithm as proposed in [Chiang+ 97], but unlike this scheme, CBRP mainly concentrates on the use of clusters in the routing process.

CBRP has the following features:

- fully distributed operation,
- less flooding traffic during the dynamic route discovery process,
- explicit exploitation of uni-directional links that would otherwise be unused,
- broken routes could be repaired locally without rediscovery,
- sub-optimal routes could be shortened as they are used.

Periodically, each node in the network sends a Hello message containing its current neighbour table. The size of the message is proportional to the degree of the node (i.e. the number of neighbours), which is around 6 to 15 for networks of average density. Some other issues associated with this protocol are as follows. CBRP’s support for uni-directional links has some problems. The current 802.11 MAC layer does not consider the existence of uni-directional links, nor does it support them. CBRP selectively makes use of those uni-directional links that could give two-way-routes to nodes that are otherwise inaccessible using only bi-directional links. There is also some delay introduced during the route discovery phase. In general this protocol has been designed for medium to large networks where Nodal mobility is not too high. This protocol is suited for networks where most of the traffic is among a small set of sender-receiver pairs compared to the possibility of N*(N-1)/2 number of pairs.

7.4.3.2 Ad-hoc On-demand Distance Vector Routing Protocol (AODV)

In [Perkins+99] an Ad-hoc On-demand Distance Vector Routing (AODV) is proposed that is an improvement on the DSDV algorithm discussed above. AODV minimizes the number of broadcasts by creating routes on-demand as opposed to DSDV that
maintains the list of all the routes. It was specifically designed for ad-hoc wireless networks and introduces minimal control overhead and minimal route acquisition latency.

If a node does not have a valid route to a destination, it initiates a route discovery process by broadcasting a route request packet. This packet contains source and destination addresses along with a sequence number. The sequence numbers ensure that the routes are loop free and to make sure that if the intermediate nodes reply to route requests, they reply with the latest information only. Each neighbour (if it is not aware of the destination) broadcasts this packet to its neighbours and this process is repeated till an intermediate node is reached that has recent route information or till it reaches the destination. Moreover, there is a lifetime associated with each route table entry that is updated whenever a route is used. If a route is not used within its lifetime, it is expired and consequently discarded, which reduces the effects of stale routes, as well as the need for route maintenance for unused routes.

When a node forwards a route request packet to its neighbours, it also records in its tables the node from which the first copy of the request came. This information is used to construct the reverse path for the route reply packet. AODV uses only symmetric links because the route reply packet follows the reverse path of the route request packet. As the route reply packet traverses back to the source, the nodes along the path enter the forward route into their tables.

If the source moves then it can reinitiate route discovery to the destination. If one of the intermediate nodes move then the moved node’s neighbour realises the link failure and sends a link failure notification to its upstream neighbours. This message propagates till it reaches the source upon which the source can reinitiate route discovery if needed.

The recent specification of AODV [Perkins+99] suggests an optimisation to AODV. This approach uses an expanding ring search to discover routes to an unknown destination. This way it is possible to search increasingly large neighbourhoods to find the destination. The TTL field in the IP header of the ‘request’ packets controls the search radius. If the route to previously known destination is needed, the prior hop-wise distance is used for the radius.

The problem with AODV is as follows. This protocol makes an assumption that there are only bi-directional links (and ignores any links that are not bi-directional) and that they all incur the same cost to use. The assumption can decrease the overall performance of the network if a large number of the links do not meet the assumption. Also, whenever an established route is broken the source must be notified and this way the source gets interrupted.

7.4.3.3 Dynamic Source Routing (DSR) Protocol

In [Johnson+99] a source routed on-demand routing protocol called Dynamic Source Routing Protocol is proposed. The two major phases of the protocol are: route discovery and route maintenance. In the Route Discovery phase, a node wishing to send a packet to a destination obtains a source route to it. This route discovery process
is only used when the source attempts to send a packet to the destination and does not already know a route to it.

In the Route Maintenance phase, a node while using a source route to a destination is able to detect that the network topology has changed such that it can no longer use its route to the destination because a link along the route no longer works. When route maintenance indicates that a source route is broken, the source can attempt to use any other route to the destination it happens to know, or it can invoke Route Discovery again to find a new route. The route maintenance phase is active only when a source is actually sending packets to a destination.

A route reply is generated when either the destination or an intermediate node with current information about the destination receives a route request packet [Johnson+96]. If the route reply is generated by the destination then it places the route record from the route request packet into the route reply packet. On the other hand, if the node generating the route reply is an intermediate node then it appends its cached route to destination to the route record of route request packet and puts that into the route reply packet.

Both route discovery and route maintenance are deployed on demand. This on-demand behaviour and lack of periodic regular advertisement messages allow the number control packets causing overheads by DSR to scale down to zero when all nodes are approximately stationary with respect to each other and all routes needed for current communication have already been discovered.

The main advantages of DSR over other popular protocols such as DSDV and AODV are as follows [Yasin+01]:

- The DSR protocol can successfully discover and forward packets over paths that contain unidirectional links.
- The DSR protocol operates entirely on demand. It does not use any periodic routing advertisement, link status sensing, or neighbour detection packets; nor does it rely on these functions from any underlying protocols in the network.

The key reason for the popularity of DSR is that it operates entirely on demand and it does not generate control overheads, as it requires no periodic activity of any kind at any level within the network. As a result of these features, the routing overhead is minimised especially when the nodes are stationary with respect to each other and this also implies that all routes needed for current communication have already been discovered. As nodes begin to move more or less communication patterns change and the routing packet overhead of DSR automatically scales to track the routes currently in use.

Having said all this, the DSR protocol has also some limitations. It relies on a flooding mechanism to do route discovery, which has an impact on most of the connected network. In their simulation, [Johnson+96] calculated the amount of network overhead versus the optimum for different network vulnerabilities. It was observed that when the network was least stable and path re-discovery is happening frequently, the packet overhead was 2.6 times optimum. The protocol also does not seem to be highly scalable because each host must keep full knowledge of the paths
that it needs to communicate over. In the worst case, if a host has to communicate with every other host on the network, the amount of information that the host must keep grows linearly with network growth. Also in a densely connected network where many hosts are within communication distance, there can be a lot of unnecessary overhead on each host due to the promiscuous listening feature. This problem also grows linearly per host in the network. Also, whenever an established route is broken the source must be notified and this way the source gets interrupted.

7.4.3.4 Temporally Ordered Routing Algorithm (TORA)

The Temporally Ordered Routing Algorithm (TORA) is a highly adaptive, efficient and scalable distributed routing algorithm based on the concept of link reversal [Park+ 97]. It is a source-initiated on-demand routing protocol for highly dynamic mobile, multihop wireless networks. This protocol finds multiple routes from a source node to a destination node. The main feature of TORA is that the control messages are localized to a very small set of nodes near the occurrence of a topological change. To achieve this, the nodes maintain routing information about adjacent nodes.

The protocol performs three basic functions: route creation, route maintenance and route Erasure. During the route creation and route maintenance phases, nodes use a height metric to establish a directed acyclic graph (DAG) rooted at the destination. Thereafter, links are assigned a direction (upstream or downstream) based on the relative height metric of neighbouring nodes. In times of mobility, the DAG route is broken and route maintenance is necessary to re-establish DAG rooted at the same destination. Tora does this by using a Reference Level. A reference level comprises of the following elements:

- Logical time of link failure.
- The unique ID of the node that defined the new reference level.
- A reflection indicator bit.

Upon the failure of last downstream link, a node generates a new reference level, which results in the propagation of that reference level by neighbouring nodes, effectively coordinating a structured reaction to the failure.

The problem with Tora is that there is a potential for oscillations to occur, especially when multiple sets of coordinating nodes are concurrently detecting partitions, erasing routes, and building new routes based on each other. Because TORA uses internodal coordination, its instability problem is similar to the "count-to-infinity" problem in distance-vector routing protocols, except that such oscillations are temporary and route convergence will ultimately occur. TORA assumes all nodes have synchronized clocks accomplished via an external time source such as the Global Positioning System (GPS). Hence it is unclear if TORA would function properly in an environment where GPS is not available or is not reliable.

7.4.3.5 Associativity Based Routing (ABR)

The Associativity Based Routing (ABR) protocol is a new approach for routing proposed in [Toh96, Toh99]. The ABR protocol defines a new metric for routing known as the degree of association stability. Associativity is related to the spatial,
temporal, and connection stability of a mobile host. Specifically, it is measured by a node’s connectivity relationship with its neighbours. It is free from the loops, deadlock, and packet duplicates. In ABR, a route is selected based on the associativity states of nodes. The routes thus selected are likely to be long-lived. All nodes generate periodic beacons to signify their existence. When a neighbouring node receives a beacon, it updates its associativity tables. For every beacon received, a node increments its associativity tick with respect to the node from which it received the beacon. Association stability means connection stability of one node with respect to another node over time and space. A high value of the associativity tick with respect to a node indicates a low state of node mobility, while a low value of the associativity tick may indicate a high state of node mobility. Associativity ticks are reset when the neighbours of a node or the node itself move out of proximity. The fundamental objective of ABR is to find longer-lived routes for ad-hoc mobile networks. The three phases of ABR are Route Discovery, Route Reconstruction (RRC) and Route Deletion.

The ABR protocol is a compromise between broadcast and point-to-point routing and uses connection-oriented packet forwarding approaches. The long-lived route determined by ABR requires fewer router reconstructions and therefore yields higher throughput. Another benefit of ABR is that it is guaranteed to be free of packet duplicates. The reason is that only the best route is marked valid, while all the other possible routes remain passive. A limitation with this protocol is that it relies heavily on the fact that each node is sending a beacon periodically. Also, the beaconing interval must be short enough to accurately reflect the spatial, temporal, and connectivity state of the mobile hosts. This beaconing requirement may result in additional power consumption. However, the experimental results in [Toh+99] reveal that the inclusion of periodic beaconing has a minute influence on the overall battery consumption.

7.4.3.6 Signal Stability Based Adaptive Routing (SSA)

The signal Stability-Based Adaptive Routing protocol (SSA) presented in [Dube+ 97] is an on-demand routing protocol. The main difference between the approaches suggested above and this approach is that SSA utilises the information available at the link layer to choose the routes. In other words the signal quality of the channel is used to determine whether portions of the topology are stable or fluctuating at any given time. Routes are only determined on demand and the protocol does not limit the rate of change of topology or suggest that all parts of the topology are equally stable. This route selection criterion has the effect of choosing routes that have "stronger" connectivity. SSA is comprised of two cooperating protocols: the Dynamic Routing Protocol (DRP) and the Static Routing Protocol (SRP).

The DRP maintains a Signal Stability Table (SST) and a Routing Table (RT). The SST stores the signal strength of neighbouring nodes obtained by periodic beacons from the link layer of each neighbouring node. Signal strength is either recorded as a strong or weak channel. All transmissions are received by DRP and processed. After updating the appropriate table entries, DRP passes the packet to SRP. SRP passes the packet up the stack if it is the intended receiver. If not, it looks up the destination in the RT and forwards the packet. If there is no entry for the destination in the RT, it initiates a route-search process to find a route. Route-request packets are forwarded to
the next hop only if they are received over strong channels and have not been previously processed (to avoid looping). The destination chooses the first arriving route-search packet to send back as it is highly likely that the packet arrived over the shortest and/or least congested path. The DRP reverses the selected route and sends a route-reply message back to the initiator of the route-request. The DRP of the nodes along the path update their RTs accordingly. Route-search packets arriving at the destination have necessarily arrived on the path of strongest signal stability because the packets arriving over a weak channel are dropped at intermediate nodes. If the source times out before receiving a reply then it changes the PREF field in the header to indicate that weak channels are acceptable, since these may be the only links over which the packet can be propagated. When a link failure is detected within the network, the intermediate nodes send an error message to the source indicating which channel has failed. The source then sends an erase message to notify all nodes of the broken link and initiates a new route-search process to find a new path to the destination.

The simulation results shown in [Dube+ 97] quantify the length and the longevity of the routes determined by SSA under various node densities and mobility rates. They also determine the relative efficacy of using signal strength and location stability as selection criteria for routing. However, one limitation of this scheme is that in cases of multiple failures, some routing messages may not reach their destinations. This may result in the existence of stale routes, but it will not cause any routing errors or loops. The stale routes will be discovered and erased by the next data packet that tries to use the invalid route. Also, SSA with location stability performs worse than that without location stability as is shown by the simulation results in [Dube+ 97]. This is because location stability introduces much stringer criteria for a link to be strongly connected. If the protocol is unable to find a strong route, route discovery process takes longer, and the route is likely to fail sooner. The other disadvantage of this scheme, similar to DSR and AODV, is that the source is always interrupted when the route is broken due to mobility, which is undesirable.

7.4.4 Remarks

To summarise, a table driven ad-hoc routing approach is similar to the connectionless approach of forwarding packets, with no regard to when and how frequently such routes are desired [Royer+ 99]. Such a scheme relies on an underlying routing table update mechanism that involves constant propagation of routing information. In contrast, a node using an on-demand protocol desires a route to the destination; it will have to wait until such a route is discovered. On the other hand, because routing information is constantly propagated and maintained in the table driven routing protocols, a route to every other node in the ad-hoc network is always available, regardless of whether or not it is needed. This feature, although useful for datagram traffic, incurs substantial signalling traffic and power consumption. Since both bandwidth and battery power are scarce resources in mobile computers, this becomes a serious limitation. Another consideration is whether a flat or hierarchical addressing scheme should be used. While flat addressing scheme may be less complicated and easier to use, there are doubts as to its scalability.
7.5 Applications of Ad-hoc Networks

Any commercially successful network application can be considered a candidate for useful deployment with nodes that can form ad-hoc networks [Perkins+ 01]. Some potential applications for which an ad-hoc network infrastructure will be beneficial are identified in this section.

- **Military Tactical Networks**: The principle requirement in a military environment is that the participating battlefield units be able to communicate anywhere and at anytime in the absence of a fixed infrastructure. The units must be capable of performing routing and other network management functions to sustain such networks.

- **Collaborative Networking**: A typical example of such a scenario is when a group of people need to have a conference or a meeting and there are no fixed infrastructure services available. If these devices are ad-hoc enabled, they could dynamically set up a network without a great deal of configuration.

- **Sensor Networks**: Recent attention has been focussed on ideas involving the possibility of coordinating activities and reports of a large collection of tiny sensor devices [Estrin+99, Kahn+99]. These communication enabled sensor nodes collectively form an ad-hoc network. Each node contains a specific sensor such as a movement or a heat sensor. When a sensor is activated, it relays the obtained information through the ad-hoc network to some central processing node where further analysis and actions can be formed. Such networks can be used in a wide variety of situations ranging from military to surveillance. Such sensor-based nodes forming ad-hoc networks have tighter constraints in terms of size, power, and processing capabilities.

- **Personal Area Networks**: The idea of a Personal Area network (PAN) is to create a very localized network populated by some network nodes such as PDA, laptop etc. that are closely associated with a single person [Perkins+01]. As an example, one can connect the PDA to a cell phone and use the phone to gain access to the Internet. Also there may be situations when several PANs may need to interact with each other. For instance, when two people interact it is highly likely that their PANs may need to interact with each other too. The protocol that is best equipped to handle communication in such an environment is Bluetooth [Haarsten+98]. It is an emerging short-range radio technology based protocol that facilitates wireless communication between PDAs.

- **Disaster Area networks and Emergency services**: In situations when existing network infrastructure is destroyed or is unavailable, ad-hoc networks can be used to quickly deploy a communication network. In a disaster area, such networks can be used to improve the communication among rescue workers and other personnel and thereby support relief operations.

- **Home Networking**: An example here would be a laptop, which is taken to and way from home to the office work environment and on business trips but yet maintaining its connection with the home PC. In ‘the home of the 21st century’ project that is being implemented in the George Washington University’s campus at Ashburn, Virginia, this idea is being taken one step further by forming a wireless network with other network-compatible devices such as motion detectors and security cameras [Yasin+ 01].
7.6 Summary

Ad-hoc networking is a networking paradigm for mobile, self-organizing networks that are interconnected through wireless interfaces and lack specialised nodes. In this chapter, we provided a general background on ad-hoc networking that included its origins, characteristics, some design challenges, routing and prominent applications for its use. As an observation, many routing protocols that are presented in this chapter may have little originality and are merely extensions of methods such as source routing, distance vector routing and link state routing that have been in existence since several years. It is acceptable if a protocol satisfies the underlying networking requirements and ultimately yields good performance. Also many challenging technical issues are identified that demand immediate attention and perhaps more research.
Chapter 8

Security Background on Ad-hoc networks

8.1 Introduction

An ad-hoc network is a collection of temporary nodes that are capable of dynamically forming a temporary network without the support of any centralized fixed infrastructure. Ad-hoc networks can be formed, merged together, or partitioned into separate networks on the fly, without necessarily relying on a fixed infrastructure to manage the operation. It is the responsibility of the participating mobile nodes to dynamically establish routing among themselves to form a temporary network. Since the participating nodes can move anytime, the end-to-end route established between any two nodes can change on a continual basis. Unlike conventional wireless networks, ad-hoc networks are dynamic and distributed in nature. Therefore this infrastructure gives rise to additional threats thereby making it difficult to achieve a uniform and acceptable level of security. This chapter begins by first examining design challenges that arise when facilitating security mechanisms in ad hoc network environment in section 8.2 In section 8.3 security threats are discussed. In section 8.4 provides a literature survey of the work done in this area. Finally, section 8.5 provides concluding remarks.

8.2 Security Design Challenges in Ad-hoc Networks

Ad-hoc networks have peculiarities not found in other traditional wireless and mobile networks. These peculiarities are highlighted in [Hubaux+ 01]. Firstly, they act independent of any provider. They are not centralised, rather they are highly cooperative such that any task is accomplished as a result of the cooperation of a group of nodes. Mobile nodes also have limited power and rely on batteries for their energy. If battery powered devices are used, the transmission power and the processor utilization will directly affect the battery lifetime [Schwingenschlogl+ 02]. The topology of such networks is highly dynamic with links and routes changing rapidly. Links are wireless with limited bandwidth and capacity. Bandwidth is a scare resource in a wireless world. This is especially true for mobile ad-hoc networks that have to cope with additional signalling information, and also have to act as relay- stations for neighbouring network nodes [Schwingenschlogl+ 02]. These links are not reliable and are susceptible errors due to transmission impairments. These peculiarities present several challenges to designing a secure infrastructure for ad-hoc network.

Ad-hoc networks rely on wireless links for communication. These links are easy targets for security breaches that can occur in the form of passive attacks (eavesdropping) or active attacks such as unauthorised modification of information. Also these links are inherently susceptible to wireless channel errors that further hamper reliable communications. Nodes that participate in an ad-hoc network often operate in a hostile environment. Such nodes lack adequate physical protection and the probability that these nodes may be compromised is very high. There is no fixed infrastructure support in an ad-hoc network to manage routing and multicasting. Rather, the participating nodes themselves act as both the end points as well as carry out routing functions. If a node is compromised it adversely affects the overall state of the routing information. Also since the topology changes dynamically, the nodes must
rely on up to date state information from its neighbours. Any malicious or a compromised node can take advantage of such a situation by propagating replay messages, sending incorrect routing information or masquerading as some other node thereby disrupting the entire network operation. This leads to a denial of service problem. Routing protocols must be robust against such potential attacks. Note that this is an example of an internal attack where one of the legitimate participating nodes is compromised. The nodes are highly mobile and may join or leave the networks frequently causing a frequent update to topology. In such situations it is essential to ensure that a node is trusted to participate in the network. Appropriate authentication and access control schemes must exist to detect an intruder in the network. This attack may be internal as mentioned above or external where nodes that do not belong to the network cause it.

8.3 Security Threats in Ad-hoc Networked Environment

Based on discussion presented in the previous section, we are now in a position to identify the security threats that are prevalent in an ad-hoc network environment.

**Denial of Service:** This threat prevents or inhibits the normal use of management or communication facilities. This is a threat against the availability of the system. Availability means that all necessary components are operable and all the necessary services are available when a user requires them. Availability has two aspects with respect to ad-hoc networks i.e., is the medium available when it is needed and are the services offered by a node available to its users when expected, in spite of attacks [Nguyen+ 00]. In the denial of service attack, the attacker can deny service to the nodes in a given area by jamming the radio frequencies they use. From the perspective of the nodes, the attack can be internal or external. In an internal attack a legitimate node behaves maliciously to subvert the communication between two or more nodes. In an external attack, an unauthorised node attempts to degrade or prevent the message flow between two nodes. Such nodes may propagate incorrect routing information, replay old routing information or even distort old routing information [Zhou+ 99]. In the former case, appropriate mechanisms and services must exist to detect and eliminate the compromised malicious node. In the latter case, appropriate authentication and authorisation services must exist to prevent an unauthorised node from participating in the network.

**Unauthorised Disclosure of Information:** In this type of attack, an attacker tries to learn the contents of the transmitted message. A confidentiality service is necessary to protect the transmitted data from such attacks. Several levels of protection can be identified. The broadest service protects all the data transmitted in the network for a period of time. Narrower forms of this service can also be defined, including protection of a single message or even specific fields within a message. In an ad-hoc network, the links are wireless and are error prone. Use of such wireless links renders an ad-hoc network susceptible to link attacks ranging from passive eavesdropping to active impersonation.

The other aspect of confidentiality is the protection of traffic flow from analysis. This requires that the attacker not be able to observe the source and destination, frequency, length, or other characteristics of the traffic. In ad-hoc networks leakage of such information could have fatal consequences. For instance, in battlefield ad-hoc
networks, traffic analysis can have devastating effects. In such situations, the routing information must also remain confidential because this information might be valuable to enemies to identify and locate their targets in a battlefield [Zhou+ 99].

Appropriate encryption (symmetric or public key) must be in place to protect the secrecy of transmitted information. These encryption mechanisms require the support of an efficient key management infrastructure. There are several issues in providing such an infrastructure for ad-hoc networks. We discuss these issues later in this section.

**Unauthorised Modification of Information:** This simply means that some portion of the legitimate message is altered, or that messages are delayed or reordered, to produce an unauthorised effect. An integrity service is necessary to guarantee that a message being transferred is never corrupted. In an ad-hoc network environment, a message could be corrupted because of benign failures, such as radio propagation impairment, or because of malicious attacks on the network [Zhou+ 99]. Integrity can be applied to a single message, selected fields within a message or to a stream of messages. When integrity is applied to a stream of messages then this process is called connection-oriented integrity. This approach ensures that messages are received as sent, with no duplication, insertion, modification, reordering, or replays. The links in ad-hoc networks are susceptible to errors and prone to active attacks, and hence the use of connection-oriented integrity can be justified. An integrity service is often implemented using Checksums and hashing functions.

**Unauthorised Access to Information:** This includes unauthorised access to resources on the network as well as within a system. This may occur in conjunction with a masquerading attack. Having successfully masqueraded as another entity, an attacker can gain access to resources, which are otherwise denied to it. An access control service provides the ability to limit and control access to host systems applications and networks, and to limit what entity might do with the information contained. In ad-hoc networks access control can involve mechanisms with which the formation of groups is controlled [Karpijoki 00]. For example, in cluster driven ad-hoc networks, an access control service must exist that determines when nodes may form, destroy, join, or leave clusters.

**Access Control** service is represented by an access control policy that is essentially a set of rules that define the conditions under which initiators may access targets. The decision to grant or deny a particular request is determined using access control rules and the access control information associated with the request. There are various approaches to implementing this access control policy. Discretionary access control (DAC) allows the restriction of access to objects based on the identity of subjects or groups of subjects. Mandatory access control (MAC) involves centralised mechanisms to control the access to objects with a formal authorisation policy. DAC and MAC are often applied together so that DAC allows the system user subjects to control access of other subjects, while MAC controls and restricts the operation of DACs in the system in general. Role Based Access Control (RBAC) applies the concept of roles within subjects.

**Masquerading:** A masquerading attack takes place when one entity pretends to be another entity. An **authentication** scheme must be in place to the recipient that a
message is from the source that it claims to be. In an ad-hoc network a node performs routing functions apart from being an end recipient for a transmission flow. A malicious node can falsify, suppress, and misdirect routing information thereby disrupting the network. Hence not only there is a need for end to end authentication between two nodes, there is also a need to provide inter node authentication prior to the exchange of control information. Hence two nodes prior to exchanging routing information must be able to authenticate each other. It must also be ensured that the communication between these two nodes is not interfered with in such a way that a third node can masquerade as one of the two legitimate parties for the purpose of unauthorized transmission and reception.

**Repudiation of Actions**: This is a threat against accountability and occurs when the sender (or the receiver) denies having sent (or received) the information. A non-repudiation service must be in place that prevents either the sender or receiver from denying a transmitted message. Thus, when a message is sent, the receiver can prove that the message was in fact sent by the alleged sender. Similarly, when a message is received, the sender can prove that the alleged receiver in fact received the message. When one node receives a false message from another node, this service allows it to accuse the other node of sending the false message and enables all the other nodes to know that the offending node is compromised [Nguyen+ 00].

**Key Management Problem**

An orthogonal problem in providing the above mentioned security services is to facilitate the provision of a key management infrastructure. Key management addresses the problem of creation, generation and distribution of keys that are necessary for the implementation of a security service. Wired and traditional wireless networks entrust the responsibility of key management to a static and trusted authority, which normally is a part of fixed infrastructure. Such an entity does not exist in an ad-hoc environment where the nodes dynamically form a network without the support and jurisdiction of any fixed authority or infrastructure. The dynamic nature of ad-hoc networks implies that nodes move in and out of the networks frequently making it necessary to change the keys in securing communications more often than in other networks. This implies that the regular key management framework used in traditional networks will not be applicable to this situation.

There are several approaches to providing a key management service. Often a public key infrastructure is deployed because of its superiority in distributing keys and in achieving integrity and non-repudiation [Zhou+ 99]. However, a public key infrastructure based scheme requires the presence of a certification authority (CA) [Gasser+ 89] for key management. The CA has a public/private key pair, with its public key known to every node, and signs certificates binding public keys to nodes. The public key system requires that the Certification Authority (CA) be available on line, to verify and revoke public key certificates. The presence of a single CA leads to bottleneck problems leading to a single point of failure. It also becomes the most obvious point of attack since a compromise of the CA compromises the entire network. Finally, such an infrastructure demands that a single reliable, trusted entity is available on-line, an assumption that appears unrealistic in an ad-hoc environment. The problem of key management is further aggravated in multicast based
communications i.e., situations involving multi-party communications. Any key management scheme deployed for such networks must have the following properties:

- Scalability: To accommodate dynamic groups of arbitrary size.
- Low computational complexity: To accommodate nodes having limited resources.
- Not relying on dedicated trusted nodes: To accommodate rapidly changing topological conditions.

In the above section, we have identified the types of threat that can rise in an ad-hoc environment. For a particular environment, one needs to determine which threats are more applicable. For instance, in group-based communication, utmost care must be taken to prevent the key management infrastructure from being compromised. In military based communication, care should be taken to protect the information flow from traffic analysis. For commercial scenarios like a wireless payment system over ad-hoc, strong authentication and authorization schemes must coexist with the confidentiality service. In most of the cases, physical attacks against the nodes must be considered. This can be a very serious security incident as the attacker can get access to hardware and software known to the network and can possibly perform successful authentication, eavesdrop messages or inject arbitrary or malicious data into the network [Schwingenschlogl+ 02]. It may be necessary for such networks to deploy intrusion detection systems to detect and respond to computer misuse.

Counteracting the relevant security threats involves provision and enforcement of appropriate security services and mechanisms as discussed above. The overall set of security measures required to counteract the identified threats constitute the security policy. The security architecture supported by this policy must be able to support a wide range of systems and applications, and consequently it is intended that it should support a wide range of security services that can used and combined in different ways to meet different security policies.

Summarising, in developing a security framework for wireless ad-hoc networks we may need to address at least the following questions:

- What are the different kinds of threats/attacks against wireless ad-hoc networks and how can we counteract them?
- What are the best-proposed methods for securing a wireless ad-hoc network?
- What is the overhead involved in both securing and routing data over an ad-hoc network using existing security mechanisms?
- How do we balance, speed, security, and reliability in the ad-hoc network environment?
- Which ad-hoc network architectures and routing protocols are specifically targeted to assist in security?
- How to determine if a node is compromised and how do you avoid it? This leads to another question: How do you establish a trusted node?
- How do you carry out key management in an ad-hoc network environment?

In developing a security service, we must address the following questions:

- What are the security information/attributes used by the service?
• What mechanisms can be used to provide the service and what are the associated rules of operation?
• What are the authorities that are involved in the management of the service and its associated mechanisms?

8.4 Previous Work on Securing Ad-hoc Networks: A Review

In this section we shall give an overview of the research work that has been done in the area of securing ad-hoc networks.

The paper by [Karpijoki 00] does a short literature study over papers on ad-hoc networking to show that many of the new generation ad-hoc networking proposals are not yet able to address the security problems they face. The author indicates that environment-specific implementations of the required approaches in implementing security in such dynamically changing networks have not yet been fully realised.

In [Stajano+ 99], the authors examined the main security issues that arise in an ad-hoc network of wireless devices. The authors argue that since the ad-hoc network environment is constrained by tight bounds on power budget and CPU cycles, and by the intermittent nature of communication, this combination makes authentication, naming and denial of service irrelevant. They identify new attacks such as sleep deprivation torture, and limitations on the acceptable primitives for cryptographic protocols. They also argue that providing confidentiality is not a challenge as opposed to providing authentication. According to the authors, if the issues relating to authentication are addressed, protecting confidentiality is simply a matter of encrypting the session using whatever keying material is available. They present the secure transient association approach that identifies the relationship between a controller and a peripheral device. This association identifies the ability of a controller to control a peripheral device during its period of ownership of the device. When the device comes under the jurisdiction of the new controller, it obeys its new master and not its previous one thus making this association transient.

The proposed solution in [Stajano+ 99] is called the resurrecting duckling model. The duckling here is the slave device while the mother duck is the master controller. The duckling will recognize as its mother the first entity that sends it a secret key on a secure channel e.g., by physical contact. This procedure is called imprinting. The duckling will always obey its mother, who tells it whom to talk to by reference to an access control list. The bond between mother and duckling is broken by death after which the duckling accepts another imprinting. Death may be caused by the mother itself, a timeout or any specific event. The whole security chain corresponds to a tree topology formed of hierarchical master-slave relationships. The root of the tree is a human being controlling all devices in its subtree.

The problem with this approach is that if one relationship is broken the relationship of the whole subtree is broken. It also requires the constant intervention of a human being for security maintenance, which may not be feasible in many cases.

In [Levijoki 00] some of the challenges in providing authentication, authorisation and accounting (AAA) in wireless ad-hoc networks are identified. This is a review paper that cites the work of [Zhou+ 99] to propose some remedial solutions.
In particular, schemes proposed in [Kong+ 01] and [Luo+ 02] have described a solution to security support in wireless ad-hoc networks. The work in [Luo+ 02] is based on the principles identified in [Kong+ 01]. In [Kong+ 01] a scalable intrusion-tolerant security solution for infrastructureless wireless mobile ad-hoc networks is described. This design was motivated by three factors: a. It is not realistic to believe any security system is unbreakable. Therefore any design has to work in the presence of such break-ins.(b) The idea is to maximise the service availability in each network locality; this is crucial for supporting ubiquitous services for mobile users. c) The solution proposed has to be fully decentralised to operate in a large-scale network.

From the cryptographic perspective, the design is based on the concepts of threshold secret sharing and secret share updates. From the system aspect, the architecture is fully distributed and localised.

Keeping these principles in mind, in their design, the authors have proposed a framework in which they have distributed the certification authority functions through a threshold secret sharing scheme, in which each entity holds a secret share and multiple entities in a local neighbourhood jointly provide complete services. They employ localised certification schemes to enable ubiquitous services. They also update the secret shares to further enhance the robustness against break ins.

There are certain drawbacks associated with this scheme. In this scheme there is a k-bounded coalition offsetting technique to enable scalable distributed certificate generation. A node \( V_i \) first locates a coalition \( B \) of \( K \) neighbours \( \langle V_1, \ldots, V_K \rangle \) and broadcasts certificate requests to them. A node \( V_j \in B \) checks its monitoring data on \( V_i \) to decide if the certification service is granted. Upon receiving \( k \) partial certificates from coalition \( B \), node \( V_i \) multiplies them together to recover its full certificate. The problem is that if any node in coalition of users fails to respond due to node failures or moving out of range, all the other partial certificates become useless. The computation of all other nodes is wasted and \( V_i \) must reinitiate the whole process from the very beginning.

The other problem is that when a node \( V_j \) receives a certificate request from \( V_i \), its records may not provide enough information on \( V_i \). It may be because the interaction between \( V_i \) and \( V_j \) does not last long enough. Moreover, \( V_i \) may not exist in \( V_j \)'s records at all if they just met. \( V_j \) has two options in this scenario. One is to serve \( V_i \)'s request, since no bad records are located. The risk is that a roaming adversary who has no hope to get a new certificate from his previous location may take advantage of this. The other option is to drop the request, since no records can demonstrate \( V_i \) well behaving. In this case, a legitimate mobile node may not be able to get a new certificate. Also the simulation results presented in [Kong+ 01] show that both centralised and hierarchical solutions incur a high delay and worse, this delay also greatly fluctuates, thus making it hard to predict some useful information, such as the future expiration time in certificate renewal and consequently the frequency of renewal.

The paper by [Luo+ 02] follows the design guidelines presented in [Kong+ 01] and makes several new contributions. They formalise a local trust model and then expand the adversary model that the system should handle. In their trust model, an entity is trusted if any \( K \) trusted entities claim so within a certain time period \( T_{cert} \). These \( K \)
entities are typically among the entity’s one-hop neighbours. In other words, this is a self-securing design approach in which multiple nodes (say K) collaboratively provide authentication services for any node in the network. This design handles two kinds of attacks: DOS attacks and node break-ins. They further propose a refined localised certification services, and develop a new scalable solution of share updates to resist more powerful adversaries. Finally they evaluate their solution using simulations.

Some deficiency is visible in the approach suggested in [Luo+ 02]. The scheme assumes that each node as at least K legitimate neighbours. This assumption is critical for certification services to be robust against the adversaries. The parameter K also determines the availability of the services. These three factors are coupled and represented by a single parameter K. This coupling effect reduces the flexibility of this scheme. In certain scenarios, these three aspects may even have conflicting goals. For instance, security may require K to be at least 10, but service availability requires K to be at most 7, while the network can only guarantee 5 legitimate neighbours. How to decouple these three aspects poses new challenges that need to be addressed. This scheme also makes an unrealistic assumption that each node is equipped with some local detection mechanism to identify misbehaving nodes among its one hop neighbourhood.

The paper by [Zhang+ 00], examine the vulnerabilities of ad-hoc networks and conclude that an ad-hoc network, due to its features of open medium, dynamic changing topology, cooperative algorithms, lack of centralised monitoring and management point, and lack of clear line of defence is particularly vulnerable to a denial of service attack. The authors argue that to build a highly secure wireless ad-hoc network, appropriate intrusion detection and response techniques must be deployed. They believe that further research is necessary to adapt intrusion detection and response techniques to an ad-hoc network environment and argue that traditional methods using encryption and authentication may reduce but not eliminate intrusions in this environment. The authors discuss a new intrusion detection technique that is distributed and cooperative and uses a statistical anomaly detection approach. They argue that the trace analysis and anomaly detection should be done locally in each node and possibly through cooperation with all nodes in the network. Further, intrusion detection should take place in all networking layers in an integrated cross layer manner. This approach is specific to intrusion detection only and is not a generic security model. Also, detection rates and performance penalties remain to be investigated.

The paper by [Schwingenschlogl+ 02], presents some algorithms that can be used as building blocks for security in ad-hoc networks. Given the constraints faced by such networks, the paper also inspects the practicability of such algorithms in ad-hoc environments. The authors introduce different algorithms that deal with secret and function sharing and RSA threshold schemes such as [Rabin 98] and [Shoup 00] and describe whether these algorithms meet the given demands of ad-hoc networks.

The papers by [Zhou+ 99, Binkley+ 01, Venkatraman+ 01, Nguyen+ 2000, Marti+ 00], Torgerson+ 01, and Papadimitratos+ 02] discuss authenticated routing schemes for ad-hoc networks.
In [Zhou+ 99] the new challenges and opportunities posed by ad-hoc networking environment are identified and new approaches to securing communication are proposed. This paper focuses on how to secure routing and how to establish a secure key management service in an ad-hoc networking environment. In particular techniques to prevent denial of service attacks from occurring in the routing process are discussed. These techniques take advantage of the redundancies in ad-hoc network topology and use diversity coding on multiple routes to tolerate both benign and Byzantine failures. According to [Zhou+ 99], this work represents the first step of their research to analyse the security threats, to understand the security mechanisms for ad-hoc networks, and to identify existing techniques, as well as to propose new mechanisms to secure ad-hoc networks. The idea suggested by this approach is to distribute trust to a set of nodes by letting them share the key management service. In particular the ability to sign certificates. This is done using threshold cryptography [Desmedt+ 90]. An \((n, t+1)\) threshold cryptography scheme allows \(n\) parties to share the ability to perform a cryptographic operation so that any \(t+1\) parties can perform this operation jointly whereas it is infeasible for at most \(t\) parties to do so. Using this scheme, the private key \(k\) of the CA is divided into \(n\) shares \((s_1, s_2, ..., s_n)\), each share being assigned to a special node. Using this share, a set of \(t+1\) special nodes is able to generate a valid certificate. As long as \(t\) or less special nodes are compromised and do not participate in generating certificates the service can operate. Even if the compromised nodes deliver incorrect data the service is able to sign certificates.

In [Zhou+ 99], the user’s identity is to be authenticated by the CA. In the presence of a single centralised CA this scheme works well. But this scheme presents some problems in a distributed CA environment. This scheme proposes a scenario where collaborative CAs are deployed as access points for security services. In this case, the user must prove his identity to all special nodes to prevent a compromised node passing on faulty information. But if the CA signs certificates without proving the identity the model cannot be used for high-value transactions. There are several other characteristics that make this approach ineffective. High mobility causes frequent route changes thus contacting the local CA in a timely fashion is non-trivial. Besides, in ad-hoc networks, the local CA may be multihops away and may also move. This not only causes complicated dynamic repartitioning of the network, but also stretches the problem of locating and tracking a local CA server. Also in this approach, multi-hop communication over the error-prone wireless channel exposes the data transmission to high loss rate. This reduces the success ratio sand increases the average service latency.

In [Venkatraman+ 01], the authors argue that effective operation of ad-hoc networks is dependant on maintaining appropriate routing information in a distributed fashion to prevent a malicious node from falsifying, suppressing or misrouting data packets. To counteract this attack, they have introduced a scheme for inter-router authentication and at the same time handle replay problems that could prevail using existing schemes. The scheme has been incorporated into the routing protocol so that security threats to routing protocols can be minimised. The paper describes the scheme using AODV routing protocol but the scheme is equally applicable to other demand driven routing protocols.

The scheme in [Venkatraman+ 01] works as follows. Whenever the route from a source A to a destination B needs to be found, the route discovery process is initiated.
For this purpose, route requests are broadcast by the source and propagated through the network. When the destination or the intermediate node with route to destination receives the route request, it sends back a route reply to the initiator of the route request. These control messages that are sent during the route discovery phase are responsible for updating the route table of the source, destination, and the intermediate nodes. This scheme provides a means for authenticating these messages to ensure that the route tables do not contain any incorrect information provided by malicious nodes. This scheme performs authentication between every pair of nodes because every intermediate node updates its routing table based on the control messages it receives. They have simulated this scheme and shown the results. This scheme has the disadvantage that it does not perform strong authentication during propagation of route requests but optionally provides integrity using message authentication codes. Unfortunately, message authentication code can only check for integrity but not prevent the replay attack.

In [Nguyen+ 00], the authors have identified the security goals, current ad-hoc routing protocols, and sources of threats to these routing protocols. They have suggested possible approaches to secure routing in ad-hoc networks with a key management framework. In particular, the authors have proposed solutions for defending internal and external attacks to mobile ad-hoc networks. For external attacks, they suggest encrypting routing messages with a private key algorithm and authenticate them using digitally signed message digests with windowed sequence numbers. Redundant paths, “aging out of false routing information, and redundancy in routing information at each node are all employed to combat against internal attacks. This is at best an overview paper of ad-hoc networks security and does not as such aim to propose a specific well defined security solution.

In [Binkley+ 01], the authors have designed an authenticated routing protocol at the link layer for ad-hoc networks. This routing protocol specifically addresses link security issues. This paper presents two key ideas. Firstly, it provides a fix to reduce the spoofing problems associated with the Address Resolution Protocol (ARP). It provides a fix by replacing ARP with a protocol where beacons are used to determine reachability. This protocol is naturally integrated with Mobile IP as agents already use a beacon system for communication. Secondly, the beacons binding IP and MAC addresses are authenticated. This prevents unauthorised entities from accessing network resources and also minimises the danger of link-layer spoofing attacks. Authenticated beacons improve the overall security of Mobile IP.

This approach does not address the threat of replay attacks. An attacker might record a link layer authenticated beacon at one foreign agent and then later retransmit it. In this case, the victim may not be present and hence unaware of the attack. The protocol described in this scheme protects routing connectivity, not user data. An attacking host might simply listen to passing traffic. This is easy to do and almost impossible to detect, and works regardless of how hosts learn each other’s MAC addresses.

The scheme in [Marti+ 00] specifically addresses the problem of misbehaving nodes. A node participating in routing may misbehave by agreeing to forward packets and then failing to do so, because it is overloaded, selfish, malicious or broken. An overloaded node lacks the CPU cycles, the buffer space, or available network bandwidth to forward packets. A selfish node is unwillingly to spend battery life,
CPU cycles, or available network bandwidth to forward packets not of direct interest to it, even though it expects others to forward packets on its behalf. A malicious node launches a denial of service attack by dropping packets. A broken node might have a software fault that prevents it from forwarding packets. To mitigate this problem the authors suggest a method of categorising nodes based on their dynamically measured behaviour. They use a watchdog that identifies misbehaving nodes and a pathrater that helps routing protocols to avoid these nodes. These two features are added as extensions to Dynamic Source Routing Algorithm (DSR) [Johnson+ 96].

The watchdog technique suggested in [Marti+ 00] has certain weaknesses. The watchdog does not detect a misbehaving node in the presence of ambiguous collisions, receiver collisions, limited transmission power, false misbehaviour and partial dropping. An ambiguous collision prevents an entity A from overhearing transmissions for another entity B. In the receiver collision problem, a node A can only tell whether B sends the packet to C, but it cannot tell if C receives it. If A sends a packet it would collide with B’s packet. B may do this for malicious reasons. But B wastes battery power and CPU time, so it is not selfish. An overloaded node would not engage in this behaviour either, since it wastes badly needed CPU time and bandwidth. Thus this second case should be a rare occurrence. Another problem can occur when nodes falsely report other nodes are misbehaving. This may result in unnecessary partitioning of the network. Also a misbehaving node that can control its transmission power can circumvent the watchdog. Finally, multiple nodes in collusion can mount a more sophisticated attack. In this case, B may forwards A’s packet to C and does not report to A if C drops the packet. It may therefore be necessary to disallow two consecutive untrusted nodes in a routed path. Finally a node can circumvent the watchdog by dropping packets at a lower rate than the watchdog’s configured minimum misbehaviour threshold. Apart from all this, the watchdog mechanism can be used to some degree to detect replay attacks but require maintaining a great deal of state information.

The Pathrater technique suggested in [Marti+ 00] is run by each node on the network and it combines the knowledge of misbehaving nodes with link reliability data to pick the route most likely to be reliable. The limitation with this technique is that since the Pathrater depends on knowing the exact path a packet has traversed, it can only be implemented on top of a source routing protocol.

The report from Sandia laboratories [Torgerson+ 01] discusses several specific threats directed at the routing data of an ad-hoc network. The authors indicate that simply appending signatures to the routing messages of currently existing routing protocols is not enough to prevent a fairly primitive adversary from inserting false information into a mobile network.

The authors have also shown that with a repeater attack, an adversary can bring down a multi-hop network even if the network has sophisticated authentication mechanisms placed on the routing messages. In this attack an adversary sets up a series of repeaters distributed throughout the physical mobile network. The repeater hears the network traffic and then transmits the traffic to all other repeaters. Then the network traffic is broadcast back in band at the new locations. Since the adversary’s repeaters have only increased the broadcast range of the mobile nodes and have not violated any of the security features, the routing protocol may behave as though nothing had
happened. The adversary's repeaters give the nodes the false belief that the diameter of the network is much smaller than that it actually is.

In [Hu+ 02] the design and evaluation of Ariadne, a new ad-hoc routing protocol is described that provides security against one compromised node and arbitrary active attackers, and relies only on efficient symmetric cryptography. Ariadne operates on demand, dynamically discovering routes between nodes only as needed. The design is based on the basic operation of the DSR protocol [Johnson+ 96]. Rather than generously applying cryptography to an existing protocol to achieve security, however, they have carefully re-designed each protocol message and its processing. These mechanisms can be applicable to securing a wide variety of routing protocols. The problem with this protocol is that it is vulnerable to an attacker that happens to be along the discovered route. In particular, there does not seem to be any means of determining whether intermediate nodes are in fact forwarding packets that they have been requested to forward.

An approach that exclusively focuses on providing a key management service to ad-hoc networks is presented in [Capkun+ 02]. Considering the extreme case where there is no central authority at all, not even in the initialisation phase, the paper proposes a fully self-organising public-key management system, in which the users generate their keys, and issue store, and distribute public key certificates. This framework can be used for securing both networking functions such as routing, and application services in mobile ad-hoc networks. This scheme works as follows. Each user is its own authority domain and is capable of issuing public key certificates to other users. Each user keeps a local certificate repository containing a subset of certificates that other users issued; key authentication is performed via a chain of certificates. The certificates are stored and distributed by the users and each user maintains a local certification repository that contains a limited number of certificates selected by the user according to an appropriate algorithm. When user u wants to verify the authenticity of the public key of user v, they merge their local certificate repositories, and u tries to find an appropriate certification chain from u to v in the merged repository.

In [Capkun+ 02], several repository construction algorithms have been proposed. The proposed algorithms take into account the characteristics of certificate graphs in a sense that the choice of certificates that are stored by each user depends on the connectivity of the user and his certificate graph neighbours. The success of this approach very much depends on the construction of the local certificate repositories and on the characteristics of certificate graphs. A certificate graph is a graph whose vertices represent public keys of the users and the edges represent public key certificates issues by the users. As in any approach that uses certificate chains, this approach assumes that trust is transitive which is often not the case in practice. The authors aim to propose a fix to this problem by looking for multiple certificate paths and to use authentication metrics.

In [Papadimitratos+ 02] the authors argue that the provision of comprehensive secure communication mandates that both route discovery and data forwarding be safeguarded. The scheme proposes a secure Routing Protocol (SRP) that counters malicious behaviour that targets the discovery of topological information. The
protection of data transmission is a separate problem to the effect that an intermittently misbehaving attacker could first comply with the route discovery to make itself part of a route, and then corrupts the in-transit data. The scheme suggests another protocol called the Secure Message Transmission protocol (SMT) that provides a flexible, end-to-end secure data forwarding scheme that naturally compliments SRP. In essence this work secures against non-colluding adversaries and does not aim to authenticate intermediate nodes that forward ‘route Requests’, and thus do not handle authorization.

SRP provides correct routing information that is factual, up-to-date, and authentic connectivity information regarding a pair of nodes that wish to communicate in a secure manner. The sole requirement is that any two such nodes must have a security association. Accordingly, SRP does not require that any of the intermediate nodes perform cryptographic operations or have a prior association with the end nodes. As a result, its end-to-end operation allows for efficient cryptographic mechanisms, such as message authentication codes. More importantly, SRP can be used in wide range of networks, without restrictive assumptions on the underlying trust, network size, and membership.

The paper by [Hubauxa+ 01] provides an overview of security problems for mobile ad-hoc networks, distinguishing threats on basic mechanisms and on security mechanisms. They then develop the idea of a self-organized public key infrastructure. Their scheme is similar to PGP in the sense that public key certificates are issued by the users. However, as opposed to PGP, they do not rely on certificate directories for the distribution of certificates. Instead, in this system, certificates are stored and distributed by the users as in [Capkun+ 02]. In fact, the work presented here is similar to [Capkun+ 02] with a more detailed description of algorithms. In [Hubauxa 01], the system replaces the centralized CA as in [Zhou+ 99] by certificate chains. Users issue certificates if they are confident about the identity, i.e., if they believe that a given public key belongs to a given user. Each user stores a list of certificates in its own repository. To obtain the certificate of another entity the requester builds a certificate chain using his repository list and implicitly trusted entity’s list until a path to an entity that has the desired certificate in its repository is found.

The approach of [Hubauxa+ 01] presents some problems. Firstly, to make the system safe, the entity’s identity has to be checked in real world before users issue certificates. Furthermore it is assumed that the certificate requester trusts each node in the recommendation chain. Finally a significant amount of computing power and time is consumed to obtain a certificate going through the certificate chain. Each node in the chain has to perform public key operations, first to check the received certificate for authentication and then to sign it before forwarding it. This cannot be done in parallel but only one after the other.

In [Asokan+ 00] a new key agreement scenario is described and various solutions to the key agreement problem in this scenario are examined. This work describes a generic two party encrypted key exchange as presented in [Bellovin+ 92] and extends it to the multi-party case. It introduces a variation of the generic protocol extending which achieves a contributory multiparty key agreement. This technique is based on the assumption that the composition of the group does not change during the session. This work also presents a fault-tolerant version of a multiparty Diffie-Hellman key
agreement protocol [Diffie+ 76], which can be of independent interest. In particular, this scheme addresses the scenario of a group of people who want to set up a secure session in a meeting room without any support infrastructure. Desirable properties of a protocol that solves this problem are:

- Secrecy: Only those entities that know an initial password are able to learn the session key.
- Contributory Key Agreement: The session key is formed of contributions from all entities. Tolerance to disruption attempts: The protocol must not be vulnerable to an attacker who is able to insert messages.

This work describes and introduces several password-based key-exchange methods that meet these requirements. A weak password is sent to the group members. Each member then contributes to part of the key and signs this data by using weak password. Finally a secure session key to set up a secure channel is derived without any central trust authority or support infrastructure.

This model works perfectly for small groups. The disadvantage is that in this scheme, authentication is done outside the IT system, e.g., the group members authenticate themselves by showing their passports or based on common knowledge. In this case, friendship or knowing each other is considered as common knowledge. This scheme is not applicable for certain scenarios. For instance, groups of people who do not know each other, or pairs of people who want to have confidential messages without the rest of the group be able to eavesdrop on the channel, are two examples. Another problem arises for large groups or groups at different locations. The secure channel to distribute the initial password is not available anymore. At this point the existing support infrastructure is required to set up a secure channel.

The scheme in [Nguyen+ 00] suggests a key management framework. To manage secret keys for the encryption needed for suggested security measures, the authors propose to implement threshold cryptography due to its distributed nature. The authors claim that by including these features in the mobile ad-hoc network routing protocol will secure it against most forms of malicious attacks.

The work proposed in [Weimerskirch+ 01] aims to provide an authentication service in an ad-hoc network environment. The authors claim that this is a core requirement to initiate a secure channel. The authors propose a new authentication model for low value transactions. It makes use of recommendations and reference to derive a trust relationship, and is scalable with respect to security. A cooperation and feedback system introduces quality and responsibility. Hardware requirements for the devices are quite low. This scheme works as follows. Alice wants to authenticate Bob. Alice first asks Bob for common knowledge, which can be a secret key and information of some recent transaction. If there is common knowledge then Bob can prove his identity. Otherwise Alice can ask Bob to give his references i.e., other devices he has done transactions with recently. Alice can then verify this transaction by asking nodes involved in this transaction about the identity of Bob. A good reference is defined as an entity that has a relationship with both Alice and Bob. Once a link between Alice’s trusted network and Bob’s reference network is found, a direct relationship between Alice and Bob can be derived. But prior to establishing a relationship Alice must evaluate trust by using an evaluation function that maps this information of all
received data of the recommendation chains to ‘yes’ and ‘no’. Once an entity is authenticated, a secure channel is initiated by exchanging a secret over the trustworthy path used to derive the relationship. A feedback technique is introduced to make the system more robust and protect it from compromised nodes. For example, if Alice receives from Dan via Cathy a positive recommendation about Bob’s identity but then gets cheated by Bob, she can inform Cathy and Dan about their wrong recommendation, and she might also put Cathy and Dan on a list of suspicious devices. Cathy and Dan can then take actions to find the security risk.

The problem with the approach suggested in [Weimerskirch+ 01] is as follows. In this scheme, for secure channel establishment, a path in the network of trusted entities replaces a shared knowledge or a trusted third party. The longer the path, the higher the probability of a malicious entity among them. Therefore much effort has to be spent on finding efficient algorithms and well-sized repository lists. Another challenge is the essential feedback system. It has to be efficient, detect fraud quickly, and take appropriate action to prevent repeated attacks.

8.5 Summary

Threats, which are prevalent in any wired networking and traditional mobile network environment, also apply to ad-hoc networks. However, these threats are further aggravated due to peculiarities within ad-hoc network infrastructure such as lack of centralized authority, dynamic topology changes, and unreliable wireless links. In this chapter we highlighted security threats that are prevalent in ad-hoc networks environment and raised some fundamental questions that we need to consider when addressing security in ad-hoc networks.

We also surveyed some prominent works for securing ad-hoc networks. It is observed that several works principally aim to provide authentication services within ad-hoc networks and out these proposals some exclusively focus on adding this service to the routing infrastructure routing protocols. Some approaches have described techniques to carry out key management within ad-hoc networks. No approach mentioned above provides a comprehensive security model for ad-hoc networks.
Chapter 9

Securing Near Term Digital Radio Ad-hoc Networks

9.1 Introduction

Cluster based control structures promote more efficient use of resources in controlling large dynamic networks. There are several different cluster based control structures and associated control algorithms that have been proposed for ad-hoc networks such as [Baker 81a] [Baker 81b] [Gerla 95, Lin 97, and Zavgren 97]. Amongst these proposals, the proposal of [Zavgren 97] has actually been deployed in large tactical networks. This proposal known as the Near Term Digital Radio (NTDR) network includes a set of networking algorithms to support mobile tactical communications. A major challenge in the design of these networks is their vulnerability to security attacks.

In this chapter, we describe the procedures and mechanisms for securing Near-Term Digital Radio (NTDR) [Zavgren 97] mobile ad-hoc networks. The chapter is organized as follows. Section 9.2 describes the NTDR scheme of ad-hoc routing. In section 9.3, we define security threats arising in this environment and identify corresponding security services necessary to develop a secure framework for mobile hosts. Section 9.4 considers the design and integration of security services in this environment. In Section 9.5 we discuss key management anonymity related issues. Finally, section 9.6 gives our conclusions.

9.2 Near-Term Digital Radio (NTDR) Ad-Hoc Networks

9.2.1 NTDR Architecture

In a NTDR network, there is a set of clusters, each containing a clusterhead, which when linked together form a routing backbone [refer to figure 13]. A cluster has a single level consisting of nodes within one hop of a clusterhead. Inter-cluster communication is restricted to clusterheads only and intra-cluster communication between nodes that are not within one hop of each other must traverse the clusterhead.

In NTDR networks, data is relayed and routed automatically between users on three separate frequency-hopping patterns. The NTDR clustering architecture is self-healing in the event of a clusterhead failure. Data can hop across up to seven nodes. Cluster members are automatically handed off between backbone clusterheads while roaming. The radio uses ROSPF (Radio Open Shortest Path First) as its routing protocol. ROSPF eliminates the HELLO protocol used by OSPF to reduce network overhead bandwidth. The topology data is derived from radio node tables stored in each NTDR node.

Election of a new clusterhead is a quick process and compensates for a clusterhead being a single point of failure. Nodes keep track of their neighbours through periodic beacon messages. All beacons contain the MAC address of the issuing node and the
lowest numbered MAC address among all nodes reachable from the issuing node. In addition to this, a clusterhead’s beacon contains:

- Clusterhead’s organizational affiliation.
- A list of cluster members.
- Quality of the link from each member as measured by the clusterhead.
- Clusterhead’s transmit power level.

Nodes receiving the beacon determine whether to affiliate with the clusterhead. Each clusterhead communicates on two different frequencies, one assigned to all clusterheads and one assigned to all members of its cluster. This prevents the intercluster and intracluster transmissions from interfering with each other. With the exception of the clusterhead, the transmission range of the cluster members is kept small to limit interference and to enable spatial reuse of frequencies among distant clusters.

A node elects itself as a clusterhead in two ways:

- It does not detect any other clusterheads in the vicinity
- It detects that it can heal a network partition i.e. if it receives beacons advertising two different partition identifiers.

The following two mechanisms limit the number of nodes that simultaneously attempt to become clusterheads following initial network deployment or subsequent node movements:
• Each node that detects one of the two above mentioned conditions for becoming a 
clusterhead waits a short random interval of time and then retests the condition; 
only if the condition remains true following the waiting period does the node 
assume the role of clusterhead.
• Each new clusterhead immediately issues beacons in quick succession 
proclaiming its status.

9.2.2 Cluster Affiliation

A node seeking cluster affiliation prefers clusters such that:
• Both the node and the clusterhead belong to the same organization.
• The signal from the clusterhead is transmitted at low power but received at high 
strength.
• The resulting cluster size is relatively small.

The clusterhead can refuse an affiliation request from a node, but once it negotiates an 
affiliation it distributes its updated cluster membership to all other clusterheads. This 
update not only informs all clusterheads of the node’s current location with respect to 
the set of clusters, to aid them in computing routes to the node; it also alerts the 
node’s previous clusterhead to the fact that the node now has a new affiliation. A 
cluster member remains affiliated with its chosen clusterhead until one of the 
following events occurs, at which time it seeks an alternate affiliation:

• The clusterhead relinquishes its role.
• The clusterhead’s beacons no longer list the member, or they indicate that the 
  quality of the link to the clusterhead has become unacceptably poor.
• The received signal strength from the clusterhead is unacceptably low.

NTDR uses CSMA/CD (Carrier Sense Multiple Access with Collision Detection) as 
its RF media access protocol (MAC) for data transmission. The sequence of events 
that occur during the transmission of data packet is shown in figure 14. The source 
radio listens for a clear channel before sending a Request to Send (RTS) burst to the 
destination radio. The RTS burst is always sent by the source radio and serves the 
purpose of requesting access to the channel. The destination radio returns a Clear to 
Send (CTS) to indicate that the channel is free and it is able to accept a data packet. 
On reception of the CTS burst from the destination radio, the source radio transmits a 
variable length data packet. The destination radio demodulates and decodes the radio 
packet through the CRC error detection procedure. If the CRC is ok, then the 
destination radio transmits a linked ACK burst to the source radio to indicate successful 
reception of the data packet.

In NTDR networks, the data packets are transmitted at a burst rate of 375 kbps at up 
to 20 watts (+43dBm) in the 225-450MHz band [Modem]. The information bits are 
coded by a ¾ rate, k=7 constraint, convolutional encoder. The resulting coded signal 
is modulated onto a Direct Sequence Spread Spectrum (DSSS) Quadrature Phase 
Shift Keying (QPSK) waveform at 500kbps. A 4 MHz transmission security 
(TRANSEC) pseudo random sequence is used for spreading the coded bits. This 
results in 16 chips of DSSS modulation per QPSK symbol.
9.2.3 Routing

In the NTDR network, the clusterheads share the responsibility for maintaining the routing backbone and hence monitor and distribute among themselves information about changes that occur in the backbone. Each clusterhead generates membership information pertaining to its cluster and link state information pertaining to its links to neighbouring clusterheads. It floods this information over the backbone and computes routes to other network nodes using this link state information. The link state includes a “resistance” metric, similar to that described by [Pursley 93], which is a measure of the interference likely to be encountered by future transmissions over the link. Clusterheads compute least resistance routes to destinations, using Dijkstra’s shortest path first (SPF) algorithm [Dijkstra 59]. They establish a forwarding mechanism based on these routes. Specifically, a clusterhead maintains information about the next hop to use for each destination, based on knowledge of the destination’s currently affiliated clusterhead and the least-resistance route to it. Whenever a clusterhead detects a change that may affect routing in the backbone (e.g., a change in cluster membership or in the value of the metric on the link to a neighbouring clusterhead), it immediately floods the updated state information to all clusterheads in the network, which can then recompute new routes reflecting this state change.

While the performance of NTDR networks depends on many variables, for a simple point-to-point connection the maximum throughput is about 250 kbps and the round trip IP packet delay is about 140 msec for large IP packet sizes. It is clear that the MAC protocols must be efficient and processed by the radio quickly to maintain high data throughputs. (Refer to figure 14).

![Figure 14: NTDR MAC Protocol](image)

9.3 Security Framework

9.3.1 Security Threats and Services

Untrusted entities must not be allowed to access messages over the wireless links. This requires a confidentiality service that prevents an eavesdropper from intercepting the communication of the legitimate parties. Provision of such a service will involve encryption mechanisms as well as key management services.
Similarly, untrusted entities must not be able to stealthily alter a message in transit. An adversary may alter information over the wireless link through active eavesdropping. The provision of integrity service will involve employing cryptographic techniques to produce checksums, which can be used to determine whether there has been any insertion, deletion, or reordering of information. Untrusted party must be not able to pretend to be another and attempt to gain privileges and access to information and resources to which it is not authorised. An authentication service must be in place, which establishes the validity of the claimed entity.

No entity should be able to deny having sent (or received) information. A non-repudiation service must be in place to thwart such an attack. In particular, this service is necessary for detection and isolation of compromised nodes.

The unreliable nature of ad-hoc networks increases the necessity of changing the keys used in securing communication more often than in other wired and wireless networks. This implies that some traditional full key distribution management protocols will not be applicable in this situation. An NTDR network is nothing but a large wireless network partitioned into smaller clusters. This implies that key management and reliable data transmission is carried out by partitioning a large multicast group into a series of subgroups having relatively few members and its own multicast address. Since the nodes move in and out of clusters quite frequently, the key management techniques used to initialise the group, add and delete members must be robust enough to accommodate these changes.

It is also necessary to prevent unauthorised access to resources and services. This requires the use of an access control service in conjunction with the authentication services. This service is not discussed in this chapter.

### 9.3.2 Clusterhead Design Choices

The ad-hoc networking environment aggravates some of the above security concerns and threats. In principle, the design choices for establishing a security framework become complicated due to the absence of a long-lived trusted authority. In this section, we will outline the assumptions that are made and look at the various design options and evaluate these options. Ad-hoc networks are on-the-fly networks with a rapidly changing topology and hence care should be taken to ensure that the proposed design does not introduce intensive computational overheads.

In the NTDR architecture, the clusterhead is centralised to the cluster and performs critical functions which are necessary for efficient working of the cluster such as maintaining the membership lists, granting/denying cluster affiliation to the nodes, routing packets within the cluster, across the clusters and so on. Therefore, it is quite natural for the clusterhead to carry out security related functions. Because the clusterhead will perform certain security related functions on behalf of the nodes, there must be a mutual trust relationship between the nodes and the clusterhead. In order to understand this trust relationship, we must first describe what security related functions fall under the jurisdiction of the clusterhead and why is it natural for us to make these assumptions.
Security Functions

A clusterhead is responsible for authenticating and authorizing the nodes before granting them the cluster affiliation status. In order to accomplish this, the clusterhead can use public or symmetric key cryptographic techniques, which are discussed below. If public key techniques are deployed, then all nodes must have an access to a trusted certification authority offline. These are discussed later.

The clusterhead may also perform admission control functions. Even though a node may be able to successfully authenticate itself to the clusterhead, the clusterhead may still refuse this affiliation request as it may be in conflict with local policy rules. Both authentication and admission control must be governed by an explicitly expressed security policy.

Within a cluster, a node only trusts the clusterhead. However, the node may also trust other members, if the clusterhead vouches for them. It is natural to trust the clusterhead, as it is the convenor for the cluster. But in some circumstances it may be natural to place trust on other members of the cluster if all of them, for instance, belong to the same organization and clusterhead vouches for them. However, this trust relationship may not cross between clusters. In other words, members of one cluster may not trust the members of another cluster due to administrative and other policy based rules.

The clusterhead therefore becomes the obvious choice to manage its node’s keys. Because the nodes trust the clusterhead, each node shares a secret master key with the clusterhead, which is known only to the clusterhead and the node. The master key negotiation may be accomplished during the initial registration time when the node undergoes the authentication process. The use of this key is described at a later stage. Intra-cluster communication must itself be protected from external nodes. The traffic within the cluster is secured by means of a group key. This group key is generated by the clusterhead who is also responsible for securely distributing this key to its cluster members. This group key must be periodically refreshed to accommodate dynamic membership changes (new members joining, old members leaving). The clusterhead being the central entity in the cluster is responsible for group key management. The clusterhead is responsible for secure inter-cluster communication. As mentioned above in section 2, the NTDR architecture restricts direct inter cluster communication to clusterheads only. Therefore all inter-cluster traffic is securely transported between the clusterheads. The sending clusterhead encrypts at one end of the pipe and the receiving clusterhead decrypts at the other end. This communication can be point to point (between two clusterheads) or point to multipoint (multicast). Therefore, each node must trust its clusterhead to securely deliver its messages to another node in a different cluster. This involves other non-security related functions such as locating the target node that must also be performed by the clusterhead. In order to carry out secure inter-cluster communication, all the clusterheads must share a pair wise key with each other as well as the group key which includes all the clusterheads. The algorithms for generating and distributing such keys are described later in section 5. There may be a situation where two nodes residing in different clusters would wish the actual messages being exchanged to be hidden from the clusterheads. Such a situation may arise in a fairly static environment where the nodes more or less belong to a single cluster but temporarily visit other clusters. In such a situation, there may be a need to hide information from the visited cluster’s clusterhead. In order to
accomplish this, the two communicating nodes must exchange a secret prior to communication. This can be achieved by deploying appropriate key management techniques. In summary, a clusterhead does the following:

- Secures intra-cluster traffic.
- Secures Inter-cluster traffic (with the cooperation of group members).
- Performs key management functions (Group key and master key management).
- Performs authentication functions.

Since any node is capable of moving into and out of the cluster, this flexibility must also hold true for the clusterhead. If the clusterhead C itself has left a cluster X and relocated itself elsewhere, then it can no longer be trusted with respect to information and traffic currently pertaining to cluster X. It may be realistic to assume that clusterheads do not change frequently and even if they do they still exist as ordinary members in the same cluster. This is because the NTDR network binds the cluster formation to some affiliation such as the organization id. This implies that a node is frequently changing its location within the cluster but inter-cluster moves are less frequent. However, it is also not realistic to rule out the possibility of the clusterheads not moving at all. For example, in a military tactical environment several clusters of NTDR networks may span the entire battlefield. Here, each node may represent a soldier on the move. The clusterhead itself may be the node associated with a soldier. The links are weak and movements of the node are random and sometimes unpredictable. Due to some unforeseen circumstances (unpredictable and random) or depending upon the military tactical requirement (predictable, not random) it may be necessary for a clusterhead to move and relocate itself in a different place, that is, a different cluster. This situation gives rise to the following two scenarios.

**Scenario One: Clusterhead C leaves the cluster X without any notification**

The technical term for notification is called a handoff process wherein the old clusterhead securely transfers the current security related state information to the new clusterhead. If this does not happen, there must be some mechanism in place by means of which one can make the security related information held by C redundant. As described in section 2, the NTDR scheme has mechanisms by means of which nodes can quickly detect that the current clusterhead is no longer available. The network itself heals quickly from this failure by immediately electing a new clusterhead. As stated before, the previous clusterhead C can no longer be trusted with the information pertaining to this cluster X. The solution to this problem is to make the information held by C redundant. This is achieved by the new clusterhead renegotiating all the security related parameters with each cluster member afresh. This results in new master keys, group keys and other security related information being generated and once this is done the cluster again reaches an equilibrium state. The complexity and the overhead associated with this scheme is proportional to the number of members in the cluster and the frequency of clusterhead movements.

**Scenario Two: Clusterhead leaves the cluster with notification**

In this case, a clusterhead C undergoes an appropriate handoff process whereby it hands over the security related state information to the new nominated clusterhead. For this to happen, the clusterhead C must be cognizant of the fact that it may need to
migrate to a new cluster in the near future and therefore decides to trigger a new election process. The NTDR architecture does not address this feature. It only describes the features that detect a clusterhead is no longer available and offers a remedial solution in the form of reinitialising the network by electing a new clusterhead. But assuming that existing circumstances indicate to the current clusterhead C that it may need to change its current cluster location, it may declare the same using a cluster broadcast. This triggers a new election process, which results in the election of a new clusterhead C\textsubscript{1}. Once C\textsubscript{1} has been elected, C can securely hand off the security related state information to C\textsubscript{1} and move on. However, this approach makes a fundamental assumption that C\textsubscript{1} is trusted to delete all the state information it maintains about this cluster before it moves on. In principle, this can be a fair assumption to make as C\textsubscript{1} was formerly a clusterhead and was honest enough to indicate to the rest of its members about its future movements and intentions. If the assumption does not hold true, then one must revert back to the mechanisms deployed in scenario one.

There are some additional issues relating to anonymity and location management. We discuss these issues in the next chapter.

9.4 Security Systems Design

In this section we discuss the process of authentication and registration of nodes and establishment of secure end-to-end communication. We make the following assumptions.

1. The scheme is based on public key cryptography. This requires a public key infrastructure in place in the form of a certification authority (CA) or a hierarchy of certification authorities for a purpose of authentication and public key distribution. A certification authority is a trusted entity that verifies the identity of a participating entity, allocates a distinguished name to it and vouches for the identity by signing a public key certificate for that entity using a private key.
2. Each node has its public and private key pair.
3. Each node is capable of storing certificates of other nodes.
4. The cluster specific secret information such as node’s secret key should not be propagated from the current home cluster to the remote cluster.
5. Nodes trust the clusterhead with respect to the assumptions made in section 10.3.2 regarding secrecy of information, key management and updation, and authorization, but not with respect to anonymity as discussed in section 10.3.2.
6. Network links are bi-directional. If a node A is able to transmit to some other node B, then B is able to transmit to A as well.

The need to opt for a public key based system over a symmetric key based approach is for the following reason. Setting up shared secret keys require authenticity and confidentiality, whereas setting up public keys only requires authenticity. Furthermore, fewer public keys are generally needed, because in a network with n nodes only n public keys are needed, and can potentially be broadcast, whereas n(n+1)/2 secret keys need to be set up in the case of pair-wise shared secret keys [Hu+02].
We shall now present our rationale for the assumptions listed above. To realise assumption (1), requires a scheme for learning, storing, and distributing public keys of all routing nodes. It also requires a guarantee that public keys indeed came from a trusted party and therefore requires the use of a certificate and a certification authority. In such cases, the certification authority keeps track of all other nodes’ public keys and distributes them when requested. All nodes must know the public key of this authority. This approach does not lend itself well to mobile ad-hoc networks since this requires the presence of an infrastructure. However, we have envisaged some scenarios wherein it is feasible to implement a public key infrastructure.

**Scenario One:** A system administrator is initially responsible for configuring and setting up the ad-hoc network. This system administrator determines the admission control into this network. This entity is responsible for generating certificates for all the participating nodes. In this case, the system administrator node acts as an offline certification authority. The public keys are usually embedded in each node during initialisation phase.

**Scenario Two:** The nodes that participate in the network belong to some legitimate organizations in a certificate hierarchy chain. These nodes have certificates issued against their name, which they put to use in the ad-hoc network. In this case, each participating node may have at maximum the legitimate list of all participants or at a minimum a list of legitimate participating organisations.

**Scenario Three:** This is similar to scenario two. In this case, the nodes prior to participating in the network have had their organizations cross certify each other and therefore are issued with certificates that have a seal of approval from all participating organization certification authorities.

**Scenario Four:** Any node can take part in the ad-hoc network. The only condition for participation is that the concerned node has a valid certificate that is issued by some certification authority (CA). It differs from scenario one in that the CA need not be the same for all nodes.

**Scenario Five:** In this scenario, a subset of participating nodes acts as the certification authority. The system administrator delegates authority to this subset of nodes or to all nodes in the network initialisation phase. This is an example of a distributed key management system where a set of trustworthy nodes shares a section of the public key of the management system. In such a scheme each trusted node keeps a record of all public keys in the network. The number of nodes needed to generate a valid signature is less than the total number of trusted nodes. Therefore, if an attacker compromises some of the nodes, a complete signature can still be validated. This solution is based on threshold cryptographic schemes as outlined in [Desmedt+ 90, Gennaro+ 96, and Gennaro+ 99]. Such schemes are vital in the detection of compromised nodes. This scheme imposes a constraint that a set of nodes must be trustworthy by default to carry out CA related functions.

**Scenario Six:** The network implements the Pretty Good Privacy (PGP) [Zimmermann 95] based public key system. In PGP, the users themselves based on their acquaintances issue certificates. Thus PGP is a self-organising system. However,
when used in large communities, PGP still relies on centrally managed certificate directories (on-line servers) for the distribution of certificates.

**Observations**

The first three scenarios require some pre-arrangement whereas scenario four can be deployed fairly quickly. However, the first three scenarios offer some level of admission control that is not visible in scenario four. Option five, based on threshold cryptography makes an assumption of the presence of trusted nodes that may not be realistic in some situations. Scenario six requires the presence on an on-line CA that does not seem to be realistic at all. Although we do not rule out a situation similar to that of PGP, in our scheme we assume that certificates are stored and distributed by participating nodes and each node maintains a certificate repository that accommodates certificates of other users. This caters to our assumption (3). This is realistic assumption considering the fact that memory today is extremely cheap and therefore it is reasonable to assume that each node has sufficient memory to store information about many other nodes.

**9.4.1 Clusterhead and Node Related Information**

The information maintained by the clusterhead and each member node is given below:

**Clusterhead**

- Clusterhead Certificate
- A master clusterhead key to secure the database. Comprise of this database compromises the security of the cluster itself.
- Master keys associated with each node.
- Cluster group key.
- Certificate of all cluster members.
- Group clusterhead key.
- Peer clusterhead key (A shared secret with a peer clusterhead). This may be optional.
- Current member list: List of all nodes that are currently active in the cluster.
- Exclude member list: Nodes that have recently been excluded from the cluster.
- Associated Timestamps and Nonces

**Node**

- Node Certificate
- Clusterhead id.
- Cluster id.
- Current member list: List of all nodes that are currently active in the cluster.
- Exclude member list: Nodes that have recently been excluded from the cluster.
- Multicast address of the cluster.
- Cluster group key
- Certificate of the clusterhead.
- List of aliases maintained by the node.
- A secret (Key encryption key) shared with the clusterhead.
The following steps provide an overview of the initialisation, registration and management/updation processes required for secure sessions in the NTDR network.

**Notation**

We use the following notations in our secure communication protocols:

- **MH**: Mobile Node Identity
- **Cid**: Cluster Identity
- **CERT<sub>X</sub>**: Public Key Certificate of Node X
- **CH<sub>X</sub>**: Clusterhead Identity
- **T/N**: Timestamp/Nonce used for freshness and correlation of messages.
- **{…}K<sub>pvt-X</sub>**: Message hashed and signed using the private key of some node X.
- **K<sub>pub-X</sub>[…]**: Message encrypted using the public key of some node X.
- **CEK<sub>K</sub>**: Cluster Encryption Key -- a group key used to encrypt all intra-cluster communication.
- **KEK<sub>X-Y</sub>**: Key Encryption Key that is a shared secret between two nodes X and Y.
- **CHG<sub>K</sub>**: Clusterhead Group key. The clusterheads of different clusters form a group using this group key.
- **CHK<sub>CHY-CHX</sub>**: A shared secret between any two clusterheads.
- **List <ID,Cert>**: A list generated by the clusterhead that contains identities and the corresponding certificates of current members of this cluster. This may be optional.
- **Locinfo<sub>X</sub>**: Location information pertaining to a node X.
- **M**: Message content being communicated to the recipient.

### 9.4.2 Authentication and Registration.

In order for any node to be a member of the cluster it must first authenticate itself to the clusterhead. Upon successful completion of the authentication process, a node is considered to be a registered member of the cluster. The authentication process is initiated when some mobile node (MH) detects via a clusterhead (CH<sub>X</sub>) beacon messages that it has entered into a new cluster range. These beacon messages are generated periodically by the clusterhead. The beacon message among other things contains the certificate of the clusterhead CH<sub>X</sub>. The process of registering with the cluster is as follows. The mobile host first obtains the public key of the Clusterhead CH<sub>X</sub> from the beacon message. It then generates the following message.

$$ (1) \quad MH \rightarrow CH_x: CERT_{MH}, MH, T_1, N_1, CH_x, \{T_1, N_1, CH_x, C_{id}\} K_{pvt-MH} $$

This message contains a timestamp T<sub>1</sub> and a nonce N<sub>1</sub> to prove its freshness and to protect against replay attacks. It includes the certificate of the mobile host CERTMH, its identity and the identity of the receiving CH<sub>X</sub>. The message also contains a signed element, which includes the nonce, timestamp, and CH<sub>X</sub> along with the cluster identity Cid that mobile host MH wishes to join. All these parameters are first signed and hashed using the private key of MH (K<sub>pvt-MH</sub>). Upon receiving this message, the
receiving $CH_x$ verifies the certificate, and then uses the public key of the MH recovered from the certificate to verify the signed element. It checks for the integrity of the message and whether $CH_x$ itself is the intended recipient. It checks the timestamp and nonce to establish whether the message is fresh.

The clusterhead validates its database to decide whether to approve or deny the request. Assuming that the request is approved, it then sends the cluster group key ($CEK_K$) along with a Key Encryption Key ($KEK_{CHx-MH}$) to the mobile host MH.

(2) $CH_x \rightarrow MH$: $CH_x$, MH, $C_{id}$, $T_2$, $N_1+1$, $K_{pub-MH}[CEK_K$, $KEK_{CHx-MH}$],
{CEK_K, KEK_{CHx-MH}, CH_x, MH $T_2$, $N_1+1}K_{pvt-CHx}$

$CH_x$ sends to MH its certificate along with its authentication token. The token includes MH’s random number $N_1+1$ proving to MH that the message is fresh and is a reply to MH’s request. The token also includes Cluster’s group key ($CEK_K$) along with a secret $KEK_{CHx-MH}$, which $CH_x$ will now share with the MH. The purpose of this secret is to isolate unicast communication with multicast and its usage will be made clear later.

The mobile host MH upon receiving this message, verifies $CH_x$’s signature to check for integrity of the message and data origin authentication. It checks whether the timestamp is current and the received random number $N_1+1$ correlates to the request it sent. It then extracts the cluster group key and the secret. Now MH can communicate securely within the cluster using the cluster group key $CEK_K$ and can also take part in secure unicast communication using the secret ($KEK_{CHx-MH}$).

9.4.3 Secure End-to-End Communication Protocols

This involves securing the end-to-end path from one mobile entity to another. Each cluster uses a different group key. The clusterheads are responsible for translating data from one key to another and routing it to other clusterheads as appropriate. If a member within a cluster sends a message to the cluster, it encrypts it under the group key of that cluster. All members of the cluster know this key, however, nobody outside this cluster knows it. To forward the message to other clusters, the clusterhead decrypts the message, re-encrypts it with the group key of the clusterheads and sends this information to all the other clusterheads in the network. In particular, we consider the following cases.

(i) Communication between two mobile hosts within a cluster.

Assume that mobile hosts MH$_1$ and MH$_2$ are in the same cluster $C_{id}$ under a clusterhead $CH_x$. Assume that MH$_1$ wishes to communicate with MH$_2$. Three options are possible.

Option (a)

Each node authenticates itself to $CH_x$ and exchanges the session keys via the $CH_x$.

(1) $MH_1 \rightarrow CH_x$: $MH_1$, $CH_x$, $T_1$, $N_1$, $KEK_{CHx-MH1}[MH_2, KS_1]$, 

205
\{MH_1, MH_2, CH_x, N_1, T_1, KS_1\}KEK_{MH_1-CH_x}

In step (1), MH_1 authenticates itself to CH_x using the secret KEK_{MH-CH_x} and requests to communicate with MH_2. The mobile host MH_2’s identity is encrypted with KEK_{MH-CH_x} to protect against disclosure to eavesdroppers. The nonce N_1 and the timestamp T_1 are used by MH_1 to provide freshness and to correlate the response to the request. The encrypted element contains MH_1’s contribution to the session key. The node MH_1 signs the signed element using the shared secret symmetric key.

(2) CH_x → MH_2: CH_x, MH_2, N_2, T_2, KEK_{CH_x-MH_2}[MH_1, MH_2, KS_1]
\{CH_x, MH_2, MH_1, N_2, T_2, KS_1\} KEK_{CH_x-MH_2}

Upon validating the request, in step 2, CH_x generates a message for MH_2, which contains the information presented by MH_1 in its request. The encrypted element is encrypted using the secret CH_x shares with MH_2, which is KEK_{CH_x-MH_2}. The message is also signed using the same secret. The node MH_2 decrypts the message using the secret key to obtain the session key component of MH_1. It also verifies the signed element by computing the one-way hash function over the given parameters by using the secret key.

(3) MH_2 → CH_x: MH_2, CH_x, T_3, N_2+1, [MH_1, MH_2, KS_2],
\{MH_1, MH_2, CH_x, T_3, KS_2, N_2+1, KS_2\} KEK_{CH_x-MH_2}

In step (3), MH_2’s response to CH_x contains MH_2’s contribution to the session key (KS_2). The encrypted element is encrypted using the secret shared between MH_2 and CH_x (KEK_{CH_x-MH_2}). The signed element is signed using the same secret. Upon receiving this message, CH_x decrypts the encrypted element using the secret to obtain KS_2. It validates the signature using the same secret. It correlates the request with the response using the nonces and checks the freshness of the message using the timestamp.

(4) CH_x → MH_1: CH_x, MH_1, T_4, N_1+1, KEK_{CH_x-MH_1}[MH_1, CertMH_2, MH_2, LocinfoMH_2, KS_2]
\{CH_x, CertMH_2, MH_2, MH_1, T_4, N_1+1, KS_2\} KEK_{CH_x-MH_1}

In step (4), CH_x provides the certificate and location details and session key contribution of MH_2 to MH_1. This protocol is self-explanatory. The node MH_1 decrypts the encrypted part of the message using the secret key and checks the integrity and data origin authentication by verifying the signature of the signed element using the same secret key.

(5) MH_1 → MH_2: CertMH_1, MH_1, MH_2, T_5, N_3, K_{pub-MH_2}[KS],
\{MH_1, MH_2, T_5, N_3, KS\}K_{pvt-MH_1}

In step (5), MH_1 generates a message for MH_2. The encrypted element contains the final session key KS, which is a combination of KS_1 and KS_2 encrypted with the public key of MH_2. The mobile node MH_2 decrypts the message using its private key to retrieve the session key. It obtains the public key of MH_1 from the certificate of MH_1 and using this public key it verifies the signature. It uses the nonce and the
timestamp to verify the freshness of the message and to correlate the request with the response.

(6) \( MH_2 \rightarrow MH_1 : MH_2, MH_1, T_6, N_3+1, KS[M], \{ MH_1, MH_2, T_6, N_3+1, M \}_{KS} \)

In step (6), \( MH_2 \) generates a response for \( MH_1 \). The encrypted element contains \( MH_2 \)'s message encrypted with the new session key \( KS \). The mobile node \( MH_1 \) decrypts the message using the session key \( KS \). It uses the same session key to verify the signature. It uses the nonce and the timestamp to verify the freshness of the message and to correlate the request with the response.

Option (b)

The nodes authenticate themselves to \( CH_X \) and exchange their certificates with each other via \( CH_X \). Having done this, they then use their certificates to exchange a session key between them. This happens only if the clusterhead as not provided the nodes with the List \(<ID, Cert>\) during authentication time or the recipient node is a new member whose certificate has not yet been propagated to the remaining nodes. This direct communication is only allowed if the two nodes are within one hop of each other.

(2) \( CH_x \rightarrow MH_1: CH_x, MH_1, T_2, N_1+1, KEK_{CH_x-MH_1}KEK_{MH_1-CH_x}[MH_1, MH_2, LocinfoMH_2 CertMH_2] \{ CH_x, MH_2, MH_1, T_2, N_1+1, LocinfoMH_2, CertMH_2 \}_{KEK_{CH_x-MH_1}} \)

In step (2), \( CH_x \) provides the certificate and location details of \( MH_2 \) to \( MH_1 \). This protocol is self-explanatory. The node \( MH_1 \) decrypts the encrypted part of the message using its private key and checks the integrity and data origin authentication by verifying the signature of the signed element using the public key of \( CH_x \). Steps (3) and (4) are similar to the steps (5) and (6) in option one except that in message (3) \( MH_1 \) supplies its session key component and in message (4) \( MH_2 \) supplies its session key component. Thereafter, a session key \( KS \) is formed using the two components for secure end-to-end communication between the two hosts.

Option (c)

The node simply encrypts the information to be conveyed using the cluster encryption key. The intended recipient also responds using the cluster encryption key. This procedure is described in (ii).

(ii) Group/unicast communication within a cluster

An alternate method of carrying out this communication is using the cluster group key \( (CEK_k[MH_1, MH_2, T_1, N_1, M]) \). This mechanism is faster but results in bandwidth wastage. All group communication within the cluster uses this procedure.

(iii) Two communicating hosts are in different clusters

Consider a mobile host \( MH_1 \) located in a cluster \( C_{id1} \) under a clusterhead \( CH_x \) wishing to communicate with a correspondent mobile host \( MH_2 \) in a cluster \( C_{id2} \) under a
clusterhead CH\textsubscript{Y}. A multicast transmission reaches only the local cluster. In order for inter-cluster transmission to work, the clusterheads of different clusters form a group and therefore share a group key called the Clusterhead Group key CHG\textsubscript{K}. The sending mobile host MH\textsubscript{1} encrypts the message using KEK\textsubscript{MH\textsubscript{1}–CH\textsubscript{x}} and sends it to the clusterhead. The clusterhead CH\textsubscript{x} decrypts the message and re-encrypts it with the secret it shares with CH\textsubscript{Y}.

**Step (1) of this protocol is the same as the step (1) in case (i), option (a).**

\begin{enumerate}
    \item CH\textsubscript{x} \rightarrow CH\textsubscript{Group}: CertCH\textsubscript{X}, CH\textsubscript{x}, T\textsubscript{2}, N\textsubscript{2}, CHG\textsubscript{K}[MH\textsubscript{2}, LocinfoMH\textsubscript{2}], \{MH\textsubscript{1}, MH\textsubscript{2}, CH\textsubscript{x}, C\textsubscript{id1}, N\textsubscript{2}, T\textsubscript{2}, m\}K\textsubscript{pvt-CHX}
\end{enumerate}

In step (2), CH\textsubscript{x} performs a search for MH\textsubscript{2} by multicasting a WHO_HAS query to the CH group. The encrypted element is encrypted under the clusterhead group key (CHG\textsubscript{K}). CH\textsubscript{x} requests for the location information pertaining to MH\textsubscript{2}. This message can be decrypted and validated only by the clusterheads. The signed element is signed using the private key of CH\textsubscript{x}. Each recipient decrypts the encrypted element using the Clusterhead Group key to determine the contents of the query. They also retrieve the public key of CH\textsubscript{x} from its certificate to verify the contents of the signed element.

\begin{enumerate}
    \item CH\textsubscript{Y} \rightarrow CH\textsubscript{x} : CertCH\textsubscript{Y}, CH\textsubscript{Y}, CH\textsubscript{x}, T\textsubscript{3}, N\textsubscript{2}+1, CHK\textsubscript{CHY-CHX}[MH\textsubscript{2}, LocinfoMH\textsubscript{2}, CertMH\textsubscript{2}, C\textsubscript{id}], \{MH\textsubscript{2}, CertMH\textsubscript{2}, CertCH\textsubscript{Y}, CH\textsubscript{Y}, LocinfoMH\textsubscript{2}, N\textsubscript{2}+1, T\textsubscript{3}\}CHK\textsubscript{CHY-CHX}
\end{enumerate}

The clusterhead CH\textsubscript{Y} receives this multicast query from CH\textsubscript{x}. It checks its database to find an entry for MH\textsubscript{2}. In step (3) CH\textsubscript{Y} sends back a unicast message encrypted using the secret (CHK\textsubscript{CHY-CHX}) it shares with CH\textsubscript{x}. The encrypted element contains the location details of MH\textsubscript{2} along with its certificate. CH\textsubscript{x} decrypts the message using the secret key to obtain location details and certificate of MH\textsubscript{2}. It retrieves the public key of CH\textsubscript{Y} from its certificate to validate the signature.

\begin{enumerate}
    \item CH\textsubscript{x} \rightarrow MH\textsubscript{1} : CH\textsubscript{x}, MH\textsubscript{1}, N\textsubscript{1}+1, T\textsubscript{4}, KEK\textsubscript{CHX-MH1}[MH\textsubscript{1}, MH\textsubscript{2}, CertMH\textsubscript{2}, LocinfoMH\textsubscript{2}, N\textsubscript{1}+1, T\textsubscript{1}], \{CH\textsubscript{Y}, MH\textsubscript{2}, MH\textsubscript{1}, N\textsubscript{2}, N\textsubscript{1}+1, CertMH\textsubscript{2}, LocinfoMH\textsubscript{2}, T\textsubscript{1}, T\textsubscript{2}, m\}KEK\textsubscript{CHX-MH1}
\end{enumerate}

In step (4), CH\textsubscript{x} sends a response to MH\textsubscript{1}’s query. The encrypted element is encrypted using the shared secret between CH\textsubscript{x} and MH\textsubscript{1}(KEK\textsubscript{CHX-MH1}) The signed element is hashed and signed using the same shared secret. The node MH\textsubscript{1} decrypts the message using this secret key to retrieve the certificate and the location information of MH\textsubscript{2}, it validates the nonces to correlate the request with the response and verifies the signed element using the secret to determine the integrity of the message. Steps (5) and (6) are similar to steps (3) and (4) in option (b) of case (i).

9.5 Key and Location Management with Anonymity for Ad-hoc Networks

9.5.1 Introduction

Providing a key management framework is challenging in NTDR ad-hoc network environment as nodes frequently move from one cluster to another. Also providing security services for multicast, such as traffic integrity, authentication, and
confidentiality, is particularly problematic since it requires securely distributing a group (session) key to each of a group's receivers. Traditionally, the key distribution function has been assigned to a central network entity, or Key Distribution Centre (KDC), but this method does not scale for ad-hoc networks as such networks do not have any pre-existing or fixed infrastructure. The absence of a centralised architecture further aggravates the problem of location management. With any traditional location management scheme, a node in a network trusts a small subset of nodes that serve as dedicated location servers and periodically updates them with its location information. Since there are no static entities that permanently reside in an ad-hoc network the notion of dedicated location servers does not scale well here. Also, a node may not like to divulge its location information with other nodes in the network. This gives birth to the notion of anonymity i.e., to withhold the identity and the movements of a node to other identities. In section 10.8.2, we first discuss key management issues related to ad-hoc networks and thereafter propose a key management framework for such a network. In sections 10.8.3 and 10.8.4 we discuss issues related to anonymity and location management and propose solution to address these problems. Finally in section 10.9 we present our concluding remarks.

9.5.2 Key Management

The principal aims of a secure multicast framework is to provide authentication and secrecy for group communication such that newly joining members must not be able to read previous group communications, and that leaving members may not follow future communications. Providing security services for multicast, such as traffic integrity, authentication, and confidentiality, is particularly problematic since it requires securely distributing a group (session) key to each of a group's receivers. Providing secure group based communication for an ad-hoc network raises an additional challenge. This is because such a network is highly dynamic with frequent changes in membership and topology. Nodes dynamically join and leave the clusters therefore any static solution not accommodating on the fly changes will not be able to preserve the security of the system. Any solution proposed must take these factors into consideration and must permit the rapid distribution of session keys for a large number of users in a manner that efficiently uses the bandwidth-limited wireless medium. In this section, we first discuss the requirements of a secure group key management service for the NTDR network and then present our solution that addresses these requirements.

9.5.2.1 NTDR Key Management Requirements

- Dynamic change in membership: Membership changes in a NTDR network are frequent. Key changes are required for all group members when a leave or join occurs. This operation may occur quite frequently. This may have high overheads if groups are large.
- Lack of a long lived central trusted authority: Unlike other traditional wired and wireless networks there is no single constant trusted authority in a NTDR network. All the nodes are mobile with limited resources therefore a change in the group membership must involve minimal involvement of nodes using minimum number of messages and computing resources. Because the group membership is dynamic, the key management technique used must guarantee that at any given
instance in time, only actual group members will be in possession of the cryptographic keys needed to participate in the group communication

These imply that any key management scheme deployed for such networks must have the following properties:

- **Scalability**: To accommodate dynamic groups of arbitrary size.
- **Low computational complexity**: To accommodate nodes having limited resources.
- **Not relying on dedicated trusted nodes**: To accommodate rapidly changing topological conditions.

Key design questions that need to be addressed are:

- Who is responsible for generating and storing the group key?
- Who is responsible for updating keys with every join or a leave operation?
- Do all group members participate to generate the group key or is it done by one single trusted entity?

An NTDR network forms ad-hoc groups in the form of clusters. We previously assumed that the clusterhead could serve as a trusted entity to coordinate packet routing and manage security for the cluster. We also discussed the possible consequences when a clusterhead itself leaves the cluster. By following this model we can make the following assumptions:

- The Clusterhead manages the cluster keys for its cluster.
- The Clusterhead mediates all communication between its cluster and other clusters (A requirement imposed by the NTDR architecture).

### 9.5.2.2 Key Generation and Distribution

There are two types of keys that are used. Cluster Encryption Key (CEK) is the key used to encrypt all cluster traffic to secure intra-cluster communication. Key Encryption Keys (KEK) is a shared secret between the clusterhead a node. KEK generated is a combination of a constant and the IP address of the node. The function can be represented as: \( KEK = f(IP) \) where:

- Function \( f \): Denotes a secure hash function, for example MD5 or SHA. The initial constants used would not be the standard ones but would be randomly chosen by the clusterhead.
- IP: Denotes the IP address of the node for which this KEK is being generated.

The advantage of this scheme is that the clusterhead need not store the list of KEKs associated with each node. It can randomly generate the KEK using the IP address of the node and the constant. The constant is independent of the IP address of the node and is used with different IP addresses to generate unique KEKs. Each of these KEKs is associated with a unique IP address. Initially, a KEK is transmitted securely to the node using the public key of the node. Only the clusterhead and the node know this
KEK. Next, the clusterhead encrypts the CEK with the KEK to deliver it securely to the node. Thereafter, the node uses CEK to encrypt /decrypt all messages.

9.5.2.3 Member Addition and Deletion

Member addition involves authenticating a new node and securely providing the key of this group. It also involves informing the other members of the group about the new member. The steps are as follows:

- Check the local access control table to determine if this node is allowed to join the cluster.
- Authenticate the node.
- If authentication is successful, add this node to the list of nodes that are part of this cluster.
- Generate the KEK for this node and securely pass this key to the node.
- Encrypt the current CEK with the KEK along with cluster details such as addresses of other nodes, lifetime associated with the cluster and so on.
- Wait for an acknowledgment from the node with a timeout value set.
- If the acknowledgment arrives within the required timeframe, the node is considered to be registered (affiliated) to the cluster. Otherwise poll the node for an extended time period and if no response is obtained, the node is considered to be dead.
- Upon successful completion, Clusterhead and the node update their databases.
- Inform other members of the cluster about the new node using the current CEK. The members update their databases.

The member deletion process involves giving all the members of the cluster except the deleted node the new group key. A member is removed from the cluster when any of the following events happen:

- The member has voluntarily left the cluster with prior notification.
- The member has voluntarily left the cluster without prior notification.
- The member is forcibly removed from the cluster. This may happen if the member node is compromised.

If any of the above-mentioned events occur, the following steps are followed:

- Place the member in the exclude-member list.
- Create a new CEK.
- Create a new status message that includes the cluster id, the new CEK and the current list of members in the cluster. Send the message to each member of the node except the node that is removed (i.e. the member placed in the excluded member list). This is accomplished by encrypting the message by KEKs of all valid members that does not include the KEK of the excluded member.
- Delete the data structure associated with the node. The clusterhead as well as other members in the cluster perform this operation.
9.5.3 Anonymity with Location Management

NTDR ad-hoc networks are short-lived and there is no fixed infrastructure administering the operation of such networks. Nodes are free to move arbitrarily, both within a network and across the networks. In such situations, it is becoming increasingly important to know where the intended mobile host is currently located and also how does communication take place (both unicast and multicast) to and from a mobile host. The fact that traditional addresses hold no meaning here further aggravates this problem. Nodes are assumed to have IP addresses that are either pre-assigned or that have been assigned in a way that is not directly related to their current position relative to the rest of network topology [Perkins 01]. This differs substantially from the way that IP addresses are assigned to nodes in the global Internet where these addresses serve as pointers to the current location of nodes. Security procedures in such networks raise an additional issue of anonymity that is considered below.

The anonymity problem is concerned with the disclosure of the identity and movement of a mobile host to other entities. Some entity apart from the mobile node itself must maintain the true identity and ephemeral identity mapping for the mobile node. In traditional mobile networks such as the IETF Mobile IP [RFC 2002], the home agent is aware of the true identity, the whereabouts and the movements of a mobile host. In such a case, the true identity of the mobile host is anonymous with respect to the foreign address and therefore the foreign agent where this node is currently located. However, in an ad-hoc network, due to the absence of a long-lived trusted authority, the above mechanism cannot be deployed. In an ad-hoc network, the mobile node is constantly on the move. But in our assumption we state that the intra-cluster movement happens on an on-going basis whereas the inter-cluster movement happens less frequently. Since the clusterhead is the only entity which carries a high degree of trust and which is responsible for inter-cluster communication and key management, it is quite natural to inform this entity about the aliases used by a node. This leads us to the key question that governs certain design related choices with respect to node anonymity: with respect to whom must the node remain anonymous?

(i) The node trusts all the clusterheads and not other individual nodes.

(ii) The node only trusts the current clusterhead (where it currently resides).

(iii) The node trusts a set of clusterheads in a distributed fashion.

In general, any of these design choices may be suitable depending on the particular context and local policy. Let us now examine these cases in some detail.

9.5.3.1 Trusted Clusterheads

If we make an assumption that a mobile node must remain anonymous with respect to other nodes but not with respect to the clusterheads, then the list of aliases maintained by the node can be maintained at a single clusterhead (similar to the home agent approach used in [RFC 2002]) or in the series of clusterheads whose clusters were visited by the node. In the former case, a single clusterhead acts as a repository that stores the binding between the true identity and the aliases of the node along with its...
location details. All the other clusterheads query this clusterhead to retrieve information related to this node. In the latter case a series of clusterheads cooperate together to determine the current location details and the identity of the node. In both these cases, the node in cooperation with the clusterheads can generate the aliases. Since the clusterhead is trusted for key management as well as for authentication and cluster maintenance, it is quite natural to make this assumption. Also, using this scheme it becomes easier to track down the current location of the mobile node. It also achieves some level of secrecy in the sense that the true identity of the node is not revealed to other nodes. This design choice can be appropriate for ad-hoc networks being deployed in the commercial environment.

However, in some situations this assumption does not provide the desired security levels. The basic issue in this design choice becomes evident when a clusterhead is compromised. If a single clusterhead acts as a repository as mentioned above, then when compromised, it can unveil the entire list of aliases held by the nodes. In the case of cooperating clusterheads, if one of the clusterheads is compromised, then the adversary only gets a piece of the entire information related to anonymity. However, this adversary by means of this compromised clusterhead can deceive other clusterheads to make them reveal the complete information concerning the node. Then the adversary can wish to choose the nodes to which it may reveal this information. Such events could cause a major problem not only in a military but also in a commercial environment. We may assume that the probability of a clusterhead getting compromised in a hostile military environment is likely to be very high and therefore in such situations we do not recommend deploying this design principle.

9.5.3.2 Trusting the Current Clusterhead

This can be a realistic assumption, as the node does not want to reveal the history of its movements to the new clusterhead. The intuitive first step in this case is to assign an alias to a mobile node when it first enters a cluster. The mobile node can generate the alias independently or in cooperation with the clusterhead. Therefore, only the clusterhead and the mobile node know this alias. The mobile node uses this alias in both intra-cluster (if the cluster policy permits the node to do so) as well as inter-cluster communications. When the mobile node detects a change in the cluster, it undergoes an authentication process with the new clusterhead. Upon successful completion of the authentication process, both cooperate to generate the alias for the mobile node. Thus no single entity maintains the alias binding for a mobile node on a permanent basis. Although this solution appears to be simplistic and robust, it suffers from certain flaws. Consider the following scenario.

Consider a node MH\textsubscript{1} under a cluster CH\textsubscript{1} using an alias A\textsubscript{1}. The node engages in both, intra-cluster and inter-cluster communication using A\textsubscript{1}. When the node changes its cluster location to CH\textsubscript{2}, it gets a new alias A\textsubscript{2} from the current clusterhead. Note that the new clusterhead CH\textsubscript{2} is not aware of the previous aliases held by the node. Only the node may have this information cached within it. The node now starts to communicate using this alias A\textsubscript{2}. Now assume that some node MH\textsubscript{2} residing in a cluster CH\textsubscript{3} was previously communicating with the node MH\textsubscript{1}, which was then disguised under the alias A\textsubscript{1}. Therefore MH\textsubscript{2} recognizes MH\textsubscript{1} through A\textsubscript{1}. However note that MH\textsubscript{2} is not aware of A\textsubscript{1}’s real identity, which is MH\textsubscript{1}. Now MH\textsubscript{2} initiates a fresh communication session with A\textsubscript{1}. From its cache it procures details of previous
session it had with $A_1$ and it naturally assumes that $A_1$ may still reside in $CH_1$. Therefore a query is discharged to clusterhead of $CH_1$ for $A_1$. Clusterhead $CH_1$ maintains no state information on $A_1$ as $A_1$ is no longer located in $C1$. The local clusterhead of $CH_3$ does a broadcast and this too fails as $A_1$ as an alias is no longer being used and hence is not valid. The broadcast of clusterhead in $CH_3$ is restricted only to the clusterhead group and does not reach the intended node ($MH_1$), as it is not a clusterhead. The sending node now assumes that $A_1$ is dead although $A_1$ is alive under a new alias $A_2$. We call this phenomenon the “surrealistic death” problem.

One possible way to avoid this problem is to introduce the notion of an identity server that acts as a repository for storing aliases used by a node. In principle, this server must not be a clusterhead because movements of a node cannot be revealed to the clusterhead, as this will cause the loss of anonymity. This solves the problem of the “surrealistic death”, as at any given instant of time there is always one entity called the identity server that is capable of performing alias to alias mapping for a node. This server must itself be secure, both by location and content. “By location” it means that the server must not be physically or logically accessible to adversaries in the network. “By content” it means that the security related information (alias mapping) held by the server is itself stored in a secure form (say encrypted by the master key of the server). This approach has a problem in that if this server is compromised the anonymity of all nodes in the NTDR network is compromised. Also this violates the principle of not having a single trusted entity in the ad-hoc network. The solution to this problem is intermediate. The identity server is not trusted with the complete alias chain of a node. Every time a node acquires a new alias it conveys this information asynchronously to the identity server. This implies that the identity server records only the aliases it receives. It neither maintains a binding between the original identity of the node and its current alias nor a binding between its previously held aliases to the current alias. The identity server itself is unable to correlate the previously held aliases to this current alias. But it may use some mathematical function over the requested alias to generate the current alias. The design principles of such an algorithm is beyond the scope of this thesis.

9.5.3.3 Trusting a Set of Clusterheads

Case (ii) assumes the presence of a static identity server. It may not always be feasible to deploy such a server in an ad-hoc network environment. In which case, the only entity that can be trusted to generate and maintain a list of aliases is the mobile node itself. The aliases can have a logical relationship such that it is possible to generate the current alias using previous aliases and some additional control information. Also, but for the originating node, no single entity must have the capability to generate the entire sequence of aliases (first to last). One such scheme is discussed in section 7.

9.5.4 Location Management Scheme
Location management is concerned with the issue of tracking a mobile user’s movements and current whereabouts. An issue related to location management is the notion of anonymity of user’s identity and movements. In other words, the problem of location management is tightly coupled with the notion of anonymity. In fact, having anonymous identities only aggravates the problems relating to location management. Therefore we have addressed these two issues as a single problem.

Implementation of anonymity is based on the notion of group signatures as described in [Camenisch 97]. This group signature scheme allows members of a group to sign messages on behalf of the group. Signatures can be verified with respect to a single group public key, but they do not reveal the identity of the signer. Furthermore, it is not possible to decide whether the same group member has issued two signatures. However, there exists a designated group manager who can, in case of a later dispute, open signatures, i.e., reveal the identity of the signer. Some of the limitations found in other such schemes are overcome here.

The length of the group’s public key and or the size of the signature does not depend on the size of the group. Otherwise this can be very problematic for large groups. To add new group members, it is not necessary to modify the group public key. This group signature scheme consists of the following four procedures:

**Setup:** A probabilistic interactive protocol between a designated group manager and members of the group. Its result consists of the group’s public key $Y_G$, the individual secret keys $X$ of the group members, and a secret administration key for the group manager.

**Sign:** A probabilistic algorithm which, on input a message $m$ and a group member’s secret key $X$, returns a signature $s$ on $m$.

**Verify:** An algorithm which, on input of a message $m$, a signature $s$, and the group’s public key $Y_G$, returns whether the signature is correct.

**Open:** On input a signature $s$ and the group manager’s secret administration key this algorithm returns the identity of the group member who issued the signature $s$ together with a proof of this fact.

We make use of this scheme to provide anonymity in our model. We introduce the notion of a trusted Certification Authority offline that is responsible for a domain consisting of a set of clusters. This CA publishes its public key $K_{pub-CA}$, the domain id (CA’s authority structure) and some other system parameters. The CA also generates a series of secret keys $K_s$’s associated with its public key. Each node prior to joining the domain, obtains the public key of the CA. Each node authenticates itself to the CA prior to obtaining the public key of the CA. Each node also obtains a set of secrets $\{X_1, X_2, X_3\}$ that serves as its secret keys $K_{S1}, K_{S2}$ and $K_{S3}$. Therefore, each node within the domain is represented in the following way:

$$MH1: MH1^{Y_1 X_1}; \ MH1^{Y_1' X_1'}; \ MH1^{Y_1'' X_1''}.$$  
$$MH2: MH2^{Y_2 X_2}; \ MH1^{Y_2' X_2'}; \ MH1^{Y_2'' X_2''}.$$  

where $Y$ component happens to be a public id computed in the following way:

$$Y = a^x$$
This is called a public element. For every corresponding secret $X_1$, $X_2$, $X_3$ a mobile node will cache a public component $Y_1$, $Y_2$, $Y_3$ corresponding to each of those secrets. The CA using its private key then signs each public element associated with a secret. This signed element is represented as:

$$V_1: \{Y_1^{X_1}\}_{K_{\text{Pvt-CA}}}$$

It is basically an Schnorr signature [Schnorr 91] on a message $m$ of the entity knowing the discrete logarithm of $Y$. This signed element is a secret associated with MH and known only to the MH. In order to prove its identity, MH must use this signed element without revealing it. This is achieved using knowledge proof schemes. The knowledge proof is constructed using discrete algorithms that allow one party to convince other parties about its knowledge of certain values, such that no useful information is leaked. These proofs also serve as signatures, and hence are called signatures of knowledge. This is represented as:

$$\text{KP}_1 \{X_1, V_1\}$$

This implies that the signer can prove that it actually holds this secret $X_1$ using the signed element $V_1$ without actually revealing the contents $X_1$ and $V_1$. This algorithm description is beyond the scope of this thesis and the interested reader is advised to read [Camenisch 97] for a more detailed explanation.

Assume that a mobile node MH enters a cluster id $\text{Cid}_1$ under the jurisdiction of a clusterhead $\text{CH}_1$. A beacon message is periodically transmitted by the clusterhead that contains amongst other things, the following information:

- The domain id.
- The cluster id.
- Identity of the clusterhead,
- Certificate of the clusterhead.

The MH upon receiving such a beacon is now aware of its whereabouts. The mobile host then generates following message to clusterhead $\text{CH}_1$.

$$\text{MH} \rightarrow \text{CH}_1: Y_1, T_1, N_1, \text{Did}, \text{Cid}_1, \text{Kpub-CH}_1\{\{Y_1, T_1, N_1\}_{K_{s1}}, \text{KP}\}$$

where:

- $T_1$: Timestamp to prove message freshness.
- $N_1$: Random number to prove message freshness.
- $\text{Did}$: Domain identifier.
- $\text{Cid}_1$: Cluster identifier.
- $\text{KP}_1\{X_1; V_1\}$: Knowledge Proof.

The alias $Y_1$ of the mobile node is signed using the secret key $K_{s1}$ ($X_1$ component) of the mobile node. The message containing alias, timestamp and nonce is also encrypted using the public key of the clusterhead $\text{CH}_1$. This ensures secrecy as the message can only be decrypted using the private key of the clusterhead. The clusterhead $\text{CH}_1$ verifies the signed element in the message by applying the public key of CA $K_{\text{pub-CA}}$. It is important to note that these signatures are distinguishable from
CA’s signatures. By using this signature scheme, the mobile node MH authenticates itself to the clusterhead CH\textsubscript{1} without revealing its real identity. Thereafter the mobile node uses this alias Y\textsubscript{1} to communicate with other nodes. When the mobile node moves to a new cluster Cid\textsubscript{2} under the clusterhead CH\textsubscript{2}: it computes the following:

\[\text{Y}_1 = a^{x_1'}\]

In other words it uses its secrets alternatively when generating its aliases in visited clusters. It then generates the following message to CH\textsubscript{2}.

\[\text{MH} \rightarrow \text{CH}_2: \text{Y}_2, \text{T}_2, \text{N}_2, \text{Did}, \text{Cid}_2, \text{Kpub-CH}_2\{\{\text{Y}_2, \text{T}_2, \text{N}_2\}\text{Ks}_2, \text{KP}_1\}\]

**KP is knowledge proof:**

\[\text{KP}_1: \{\text{X}_1': \text{Y}_1' = g^{x_1'}\}\]

**Location Management**

We make the following assumptions:

1. If any two mobile nodes are to have a secure session between them, they must negotiate a session key.
2. If the nodes are communicating for the first time, they negotiate a session key using their alias certificates.
3. This session key has a lifetime L. If the nodes wish to correspond with each other, and the lifetime L of the key has expired, then they need to renegotiate a new key.
4. In our scenario, a mobile node MH has three aliases Y\textsubscript{1}, Y\textsubscript{1'} and Y\textsubscript{1''} associated with its real identity.

Assume that this MH is currently using alias Y\textsubscript{1'}. It resides in Cid\textsubscript{4} under CH\textsubscript{4}. The mobile node uses Y\textsubscript{1'} to communicate with other nodes. Also assume that some node MH\textsubscript{1} under CH\textsubscript{5} that was previously communicating with MH then disguised as Y\textsubscript{1}. The node MH\textsubscript{1} now generates a message for Y\textsubscript{1}. This node (MH\textsubscript{1}) is unaware that MH (alias Y\textsubscript{1}) now exists as MH (Y\textsubscript{1'}) and is now located in Cid\textsubscript{4}. The query is sent to the current CH of MH\textsubscript{1}. A unique ID identifies this query and is carried as part of the query. In this case let the query id be Qid. The clusterhead CH\textsubscript{5} validates its database to determine if Y\textsubscript{1} is currently residing in its cluster. If the answer is non-affirmative, then CH\textsubscript{5} sends a *WHOHAS* query to all clusterheads. This message contains the original Qid of MH\textsubscript{1}. CH\textsubscript{5} also maintains a binding table that binds the Qid with the identity of the initiator MH\textsubscript{1}. This query is received by CH\textsubscript{4}, which then broadcasts this message within its cluster. Y\textsubscript{1'} under CH\textsubscript{4} receives this query. CH\textsubscript{4} also maintains a binding template that binds Qid with CH\textsubscript{5}. Y\textsubscript{1'}'s goal is twofold. It must not reveal its current identity (Y\textsubscript{1'}) to MH\textsubscript{1} and it must also conceal its previously used identity (Y\textsubscript{1}) from the current clusterhead. Y\textsubscript{1'} constructs the following response M for MH\textsubscript{1}, and encrypts the message using the shared secret with MH\textsubscript{1}. The message structure is as follows:

\[\text{KMH}_1-Y_1\{\text{Qid, MH}_5, \text{Y}_1, \text{M}\}\]
This is the actual message token represented as T.

It then encapsulates this message using an outer envelope. This entire message element (envelope + original message) is sent to Yi’s current clusterhead (CH4). The format of the message element is as follows:

\[ Y_{1}, \text{CH}_{4}, \text{Qid}, \text{T} \]

Based on Qid, CH4 then generates the following message element for CH5.

\[ \text{CH}_{4}, \text{CH}_{5}, \text{Qid}, \text{T} \]

The message of CH4 arrives at CH5. CH5 forwards the original encrypted token along with the Qid to MH5:

\[ \text{Qid}, \text{T} \]

Using Qid as an index, MH5 retrieves the shared session key that it had previously established with Y1. It then decrypts the message using this key.

With this scheme, the messages are still received by MH even though its alias has changed several times. No entity but for MH knows its own history of aliases. It is also not possible for any other entity to keep a complete track record of MH’s movements. This method also obviates the need to have another trusted entity such as the identity server to keep track of the aliases.

9.6 Summary

Providing security for mobile ad-hoc networks is a challenge that has not been adequately met so far. In this chapter we have looked into the challenges that an NTDR ad-hoc network poses in terms of security. We proposed a security infrastructure for this network that provides secure intra-cluster and inter-cluster communication. The routing information is propagated frequently within a cluster and across the clusters to keep the routing state fresh and consistent. An adversary may tamper with the routing topology information being exchanged between two nodes to bring the network into an inconsistent state thereby causing an attack on availability (denial of service attack). The protocols to carry out secure exchange of routing information between the nodes as well as secure partitioning of ad-hoc networks need to be addressed. A security framework must be developed to address the requirements for private clusters (anonymous clusters). Providing a key management framework for an ad-hoc network can be challenging as topology changes frequently and there is no centralised administrative authority to rely on. This also aggravates the problems related to secrecy of node movements. Taking into consideration the peculiarities of an ad-hoc network, in this chapter we have proposed schemes to facilitate dynamic key management, and location management with anonymity.
Chapter 10

Securing the Ad-hoc On Demand Distance Vector Protocol (AODV)

10.1 Introduction

Ad-hoc routing protocols are challenging to design and the need for facilitating security services further complicates matters. In such networks topology changes quickly due to rapid node mobility and therefore wired routing protocols do not scale well for such networks. So far the proposals for routing in ad-hoc networks offer no security mechanisms at all, or have only partial solutions for protecting the routing facilities [Venkatraman+ 01].

In this chapter we present a scheme for providing security services for routing of control messages in an ad-hoc network. As mentioned in the previous chapter, there are two classes of ad-hoc routing protocols namely on demand and table driven. Our focus is on on-demand routing protocols for ad-hoc networks, specifically the Ad-hoc On Demand Distance Vector Routing protocol. To summarise, in an on-demand scenario, a node attempts to discover a route to some destination only when it has a packet to send to that destination. On the other hand, in table driven routing protocols consistent and up-to-date routing information to all nodes is maintained at each node. It has been demonstrated that on-demand routing protocols perform better with significantly less overheads than table driven routing protocols in many situations [Broch+ 98, Johansson+ 99, Maltz+ 99, and Perkins+ 99]. This is because on-demand protocols are able to react quickly to changes that may occur in node connectivity and at the same time are capable of reducing or in some cases eliminating overhead in the network segments where the changes are less frequent [Hu+ 02].

We have chosen AODV as our underlying ad-hoc routing algorithm since it has several key characteristics. It drastically reduces broadcasts resulting from a link break. Also movements of nodes only have local effects and do not impact the network on a global scale [Perkins+ 99]. It is capable of supporting both unicast and multicast routing. The experiments conducted have proven that AODV scales well to large node populations revealing reduced delay and efficient packet delivery measurements [Perkins+ 99]. These encouraging results have led to the development of several AODV test beds such as those at NIST, University of California at Santa Barbara (UCSB), and Uppsala University. It should be noted that although our scheme is designed for the AODV protocol, it is also equally applicable to other on-demand routing protocols.

This chapter is organised as follows. In section 10.2, we summarise the basic operation of AODV protocol. In section 10.3, we identify security issues that are prevalent in AODV environment. This discussion clearly reveals the goals that need to be targeted when securing such protocols. In section 10.4, we present our security framework wherein we outline our assumptions and present our protocols that extends the basic AODV model to offer the desired security services. In section 10.5, we discuss issues related to securing route and reply AODV messages and explain the formats of these messages with security related extensions. In section 10.6, we discuss issues relating to secure maintenance of the AODV routing protocol. In section 10.7 we summarise our observations. In section 10.8 we discuss security related extensions.
to AODV messages. In section 10.9, we compare our scheme with other such schemes. Finally in section 10.10, we present our concluding remarks.

10.2 Basic Operation of AODV

AODV creates routes only when they are required. When a sender does not already have a route to a destination it creates one by broadcasting a route request (RREQ). It sets a timer, to avoid unbounded waiting for a reply (if the timer expires, the source may re-broadcast the request). Nodes receiving this request will generate a reply (RREP) if they are the destination or possess a current route to the destination. If neither is the case they will broadcast the RREQ (unless they have already processed that particular request). RREP messages are unicast. This process is illustrated in figure 15.

AODV uses sequence numbers to avoid loops in routes. A RREQ includes the source’s latest sequence number and the latest sequence number of the destination that it knows. It also includes a broadcast ID and the source node’s IP address (to identify the particular route request). Sequence numbers are used by nodes in determining whether to send a RREP in response to a RREQ. If a receiving node holds a sequence number for the destination equal to or greater than that in the RREQ than it generates a RREP. Each forwarding node must keep the source IP address and broadcast IP entry for each RREQ it receives, for a specified length of time.

Nodes receiving a RREQ also use it to update their information for the source node. The node sets up a reverse route entry for the source node in its route table. This reverse route entry contains the source node’s IP address and sequence number as well as the number of hops to the source node and the IP address of the neighbour from which the RREQ was received. This neighbour is the next hop toward the source.
node from the node processing the RREQ. In this way, the node knows how to forward a RREP to the source if one is received later [Perkins+ 99].

To minimise the impact of flooding RREQ messages for large networks, an expanding ring search technique is suggested in [Perkins+ 99] that allows a search for increasingly large areas of the network if the destination is not found without impacting the bandwidth. To implement the expanding ring search technique the time to live (TTL) field is set to an initial value by using ttl_start parameter. If no reply is received from the discovery period, the next RREQ is broadcast with a TTL value increased by an increment value. This process is repeated till a threshold value is reached beyond which the RREQ is broadcast across the entire network up to rreq_retries more times.

The two cases that need to be considered when responding to the RREQ message are whether the response is initiated by the intended destination itself or by an intermediate node. If the destination is responding then it must place its current sequence number in the packet, initialise the hop count to zero and also place a lifetime field that determines how long is this route valid. On the other hand, if an intermediate node is responding, it places its record of the destination’s sequence number in the packet, sets the hop count equal to its distance from the destination, and calculates the amount of time for which its route table entry is valid. In either case the initiator of the RREP unicasts it toward the source node, using the node from which it received the RREQ as the next hop.

As the RREP propagates back to the source, the intermediate nodes are required to set up forward pointers to the destination. Each such forward pointer entry contains the IP address of the destination, the IP address of the neighbour from which the RREP arrived, the sequence number for the destination and the hop-count, or the distance to the destination. Each entry is also associated with a lifetime, which is set to the lifetime contained in the RREP. Each time a route is used, its lifetime is updated. If the route is not used within the specified lifetime, it is deleted. After processing the RREP, the node forwards it toward the source. If a node receives multiple RREP messages associated with the same destination then it will forward them only if the later RREP has either a greater destination sequence number or a smaller hop count than previously sent RREP. The source node can begin data transmission as soon as the first RREP is received and can later update its routing information if it discovers a better route.

As long as the route remains active, it will continue to be maintained. A route is considered active as long as there are data packets periodically travelling from the source to the destination along that path. Once the source stops sending data packets, the links will time out and eventually be deleted from the intermediate node routing tables. If a link break occurs while the route is active, the node upstream of the break propagates a route error (RERR) message to the source node to inform it of the now unreachable destination(s). After receiving the RERR, if the source node still desires the route, it can reinitiate route discovery. Also, if a node receives a data packet destined for a node for which it does not have an active route, it creates a RERR message for the destination node. It then broadcasts the RERR message as previously described. In this way, the node without the route that is receiving the data packets
can inform its upstream neighbour that it should stop sending the data packets meant for this destination.

Neighbourhood information is obtained from periodic hello broadcasts sent by neighbouring nodes. These messages help in maintaining local connectivity of a node. This message is a special unsolicited \textit{RREP} that contains the node’s IP address and the current sequence number. It is prevented from being rebroadcast outside the neighbourhood of the node by restricting the \textit{ttl} value in the packet to 1. The failure to receive any transmission from a neighbour in the time defined by the periodic transmission of several hello messages is an indication that the local connectivity has changed and that the route information for this neighbour should be updated.

\section*{10.3 Security Issues in AODV}

The attacks that can occur in any network may be either passive or active. Passive attacks have to do with eavesdropping on or monitoring transmissions. This type of attack is a threat against the privacy or anonymity of communication. Our aim in this thesis is to protect the functioning of the routing protocol of AODV. Hence we believe that passive attacks are not an issue here and so choose to ignore such attacks. In contrast, in an active attack the attacker can modify the contents of transmitted data and might also inject packets into the network. These attacks can manifest themselves in a variety of ways in an AODV environment, such as altered packets, deleted packets or additional packets. Attacks can be further classified as internal or external attacks. Internal attacks originate at nodes that belong to the network. They may occur due a node behaving maliciously because it has been compromised. On the other hand, external attacks are instigated by nodes that do not belong to the network. Our model proposes schemes to protect against both internal and external attacks, as noted below.

An attacker can act to deny resources or services to entities, which are authorised to use them. This attack is a denial of service attack and can manifest itself in a variety of ways. For instance, the attacker can over-flood a node in the network by initiating multiple false messages (RREQ/RREP) thereby overwhelming the receiver/network. The network is flooded with wasteful packets thereby depleting the bandwidth. The attacker can also impersonate another node and propagate false control messages such as route requests and replies, causing inconsistencies in the routing tables. By sending such forged packets the attacker could, for example, route all packets for some destination to itself and then discard them. The attacker may even attempt to route such packets to a network segment where the intended destination does not reside [Hu+ 02]. The attacker could also replay such messages [Venkatraman+ 01]. Another type of attack that may frequently occur is called a rushing attack. Such attacks are specifically targeted at on-demand routing protocols that use duplicate suppression at each node [Hu+ 02a]. In such attacks, the attacker disseminates RREQ messages quickly throughout the network, suppressing any later legitimate RREqs as nodes drop them due to the duplicate suppression. These types of attacks are a threat against the availability of the network as the attacker can deny to the nodes in a given area of the network.

Such attacks place an additional overhead on the limited resources available at the node and in the network in terms of battery power, memory and buffer, and
bandwidth. While there is no specific security service to counteract the denial of service attack, the presence of an authentication service along with the other services mentioned below help mitigate ill effects due to such attacks. In principle an authentication service ensures that communication from one node to another is genuine and helps to avoid attacks due to impersonation.

An attacker can also maliciously alter the contents of control messages, which could result in incorrect route information being propagated. For example, if the fields in the \textit{RREQ} and \textit{RREP} messages are altered in transit, it could lead to disastrous results. This is also called a routing disruption attack. An example of such an attack can be that the attacker modifies the contents of the packet in an unauthorised way causing these packets to create a routing loop. The packets never arrive at intended destinations and also consume resources such as memory and computation power [Hu+ 02]. An integrity service is therefore necessary to protect the nodes from maliciously altered messages. The presence of such a service will assure the receiver that the message sent from a genuine source has not been altered in transit.

A non-repudiation service ensures that the origin of the message is legitimate. When an attacker sends a false message to a target receiver, the presence of a non-repudiation service allows the receiver and all other nodes to know that the sender is the offender and has indeed been compromised.

Considering the above we have determined a number of basic requirements for security in AODV routing.

It should be possible to identify the source of a message. This will make it easier to identify the source of false messages and help mitigate the effects of denial of service attacks. This applies to all AODV messages (\textit{RREQ}, \textit{RREP}, \textit{RERR}).

It should be possible for the initiator of a \textit{RREQ} to discover all the nodes that make up a path when it receives a \textit{RREP}. This will allow it to ignore a path if it considers one or more of the nodes on the path untrustworthy. A node may determine that another node is untrustworthy based on the second node’s past behaviour. The mechanics of this are outside the scope of this thesis.

It should be possible to detect if a message has been altered in transit. This will prevent malicious nodes altering the contents of messages.

10.4 Security Framework for AODV

In this section we propose a security framework for AODV to meet the requirements given in the previous section. We make the following assumptions:

1. The scheme is based on public key cryptography. This requires that a public key infrastructure be in place in the form of a certification authority (CA) or a hierarchy of certification authorities for the purpose of authentication and public key distribution. A CA is a trusted entity that verifies the identity of a participating entity, allocates a distinguished name to it and vouches for the identity by signing a public key certificate for that entity using a private key.
2. Each node has a public and private key pair.
3. Each node is capable of storing its own certificate and, as required, those of other nodes.
4. Network links are bi-directional. If a node A is able to transmit to some other node B, then B is able to transmit to A as well.

5. Security services are implemented by extending existing control messages of the AODV protocol. There are no changes to the protocol operation itself but each node now performs additional, security related functions, when AODV messages are exchanged.

We opt for a public key based system over a symmetric key based approach for the following reason. Setting up shared secret keys requires pre-existing confidentiality, whereas a public key system does not. Furthermore, fewer public keys then secret keys are generally needed, because in a network with \( n \) nodes only \( n \) public keys are needed, and can potentially be broadcast, whereas \( n (n+1)/2 \) secret keys need to be set up in the case of pair-wise shared secret keys [Hu+ 02].

10.5 Secure Route Request and Reply

In outline our protocol extends AODV as follows:

1. **RREQ** messages securely identify the originator of the message and the most recent broadcasting node (if different). This allows nodes to be confident of where any resulting **RREP** message should be sent. A node may also decide to ignore a **RREQ** if it considers the originator or most recent broadcaster untrustworthy.

2. **RREP** messages securely identify the immediate sender and destination of the message, the ultimate destination (i.e. the originator of the corresponding **RREQ**) and all the nodes on the route from the destination requested in the **RREQ** back to the current sender. This will provide the originator of the **RREQ** with a secure route list. The originator will then be able to judge the route on the trustworthiness of its composite nodes (if known) and/or make future judgements of their trustworthiness based on their performance in this route.

3. **RRER** messages securely identify their originating node. A sender may then decide, based, on this identification, whether the message is trustworthy or use the accuracy of this message to make future decisions about the trustworthiness of its sender.

These extensions require, amongst other issues, that no attacker is able to tamper with (remove from, change or insert) a node identity in any of the messages.

The essence of our modifications is to allow recipients of messages to identify the senders and to then use that identity information in either or both of deciding whether to trust the message or to base later trust of the sender on their subsequent behaviour. In this thesis, we provide the mechanisms for determining identity. Mechanisms for making and recording decisions about trust are outside the scope of this thesis.

We use the following notation to describe our security protocols and cryptographic operations:

- **RREQ-U**, **RREP-U** and **RRER-U** are the existing (unsecured) versions of the AODV protocol messages. **RREQ-U\(x\)**, **RREP-U\(x\)** and **RRER-U\(x\)** are messages in the original format sent by node \(x\).
The route discovery process in AODV begins when the source node floods the network with \textit{RREQ} message. In response to this request, the target node or some intermediate node must respond with a \textit{RREP} message. Assume that a source \textit{S} is trying to discover a route to a destination node \textit{D} and that such a route exists, with intermediate nodes \textit{B} and \textit{C}. The protocol steps are as follows:

\begin{enumerate}
  \item \textit{S} \rightarrow \textbf{Broadcast}: \textit{RREQ-U}_S, L_S, T_S, \textit{Cert}_S, \textit{Sig}_S
\end{enumerate}

where

\begin{align*}
\textit{Sig}_S : \{\textit{RREQ-U}_S, L_S, T_S\} \textit{SK}_S
\end{align*}

In this message the sending node must authenticate itself to other nodes when passing its request to locate a target destination. The source \textit{S} achieves this by broadcasting the \textit{RREQ} with the following security extensions. It contains a lifetime \textit{L}_S that indicates how long this request is valid. If a target node receives this request message and finds that the period \textit{L}_S has expired it must discard this message. A timestamp \textit{T}_S together with the \textit{RREQ ID} in the original message format will indicate the freshness of the message and help prevent replay attacks. The message content is signed under the private key of the sender so that receiving nodes can verify the message contents. The sender’s certificate is included for the benefit of those nodes that do not already have that certificate. The sender’s identity is not separately included in the signed part as it is part of \textit{RREQ-U}_S.

Upon receiving such a message, the next hop node \textit{B} validates the signature of the signed element using the public key of \textit{S} contained in \textit{S}’s certificate (if necessary, after validating the certificate itself). \textit{B} then checks for replays, using the \textit{RREQ ID} and timestamp and validates the lifetime of the message to determine whether it has expired or not. If any of these tests fail then \textit{B} must discard this message. Otherwise, \textit{B} rebroadcasts the \textit{RREQ} as follows:

\begin{enumerate}
  \item \textit{B} \rightarrow \textbf{Broadcast}: \textit{RREQ-U}_B, L_S, T_S, \textit{Cert}_S, \textit{Sig}_S, \textit{B}, \textit{Cert}_B, \textit{Sig}_B
\end{enumerate}

where

\begin{align*}
\textit{Sig}_B : \{\textit{RREQ-U}_B, L_S, T_S, \textit{B}\} \textit{SK}_B
\end{align*}

The objective of message (2) is to allow nodes that receive this \textit{RREQ} message to securely know where to return any reply and to identify replayed messages. \textit{B} adds to the initial message its identify, certificate and signature. Any node receiving this
message will then be assured that \( B \) did send such a message. Replays may be identified by the timestamp and \( RREQ \, ID \) in the initial message together with the intermediate node identity. If the node receives the same initial request from two different intermediate nodes (which is possible as many different routes may exist in the network) it discards the later arrivals, according to the existing AODV definition. Note that the only difference between \( RREQ-U_S \) and \( RREQ-U_B \) is in the value of the hop-count field (0 in \( RREQ-U_S \), higher for intermediate nodes). Even though \( RREQ-U_S \) is not re-broadcast intermediate nodes will thus be able to recreate it for signature checking purposes. Any further intermediate nodes will rebroadcast the request, after updating the hop-count field and replacing \( B \)'s identity, certificate and signature with their own.

This process is repeated until the \( RREQ \) message reaches the intended destination (in our example \( D \)) or a node with an active route to the destination. The node validates the originator \( S \) and the immediate re-broadcaster (in our example \( C \)) by verifying the respective signed elements. It also verifies if the lifetime period \( L_S \) is still valid and checks for replays or if the \( RREQ \) has already been processed. If all these tests are passed and the node in question is the final destination then it generates the following response. The response message \( RREP \) takes the path of the \( RREQ \) in the reverse direction back towards the source (in our examples \( D, C, B, A \)).

\( (3) \, D \rightarrow C: \, RREP-U_D, \, RREQ_ID, \, L_D, \, T_D, \, N_D, \, route\_list, \, D, \, Cert_D, \, Sig_D \)

where

\[ \text{Sig}_D: \{ RREP-U_D, \, RREQ_ID, \, L_D, \, T_D, \, N_D, \, route\_list \} \, S_{K-D} \]

\[ \text{route\_list: D, C} \]

The objective of message (3) is for \( D \) to authenticate itself to the previous hop (here \( C \)) as well as, eventually, to the source and to begin building up the route list. The signature of \( D \) binds the reply to the initial request and also identifies \( C \) as the next node on the route. This will, in part, enable \( S \) to learn the identity of all nodes along the route. The lifetime \( (L_D) \) gives a validity period for the reply. If a receiving node upon examining this response message determines that the lifetime period \( L_D \) has expired then it must discard this message. As receiving nodes may use this reply to form an active route for the purposes of subsequent requests, the lifetime \( L_D \) could be relatively long. The timestamp \( (T_D) \) and nonce \( (N_D) \) enable receiving nodes to detect replays of the reply.

Note that the route list contains the node sending the message, and the next node back along the reverse route to the original sender. This allows the sender to ensure that the \( RREQ \) took the stated route back to it, by comparing the certificates of each node to their identity given in the previous nodes signed element.

Upon receiving such a message, a node intermediate between the destination and source (here \( C \) and \( B \)) checks the signature and for replays. \( C \) obtains the public key of \( D \) from the certificate of \( D \). It can use the contents of the message to determine (from the information it stored when processing the \( RREQ \)) which request this is a reply to and to which node the reply should now be sent (in this case \( B \)). It then generates the following message to be sent to the previous hop towards the source.
(4) C → B: RREP-U_D, RREQ_ID, L_D, T_D, N_D, route_list, Cert_D, Sig_D, Cert_C, Sig_C

where

\[ \text{Sig}_C: \{ \text{RREQ\_ID, L_D, T_D, N_D, route\_list} \} S_{K\cdot C} \]

\[ \text{route\_list: D, C, B} \]

The objective of the message (4) is for C to authenticate itself to its previous hop B and, eventually, to the source. C adds the next node (here B) to the route list and adds its certificate and signature to those of D. Note that if C had possessed an active route to D when it received the RREQ it could have created this message at the time, as nodes store the route lists and signatures associated with active routes\(^3\). C sends the message to B.

Note that intermediate nodes (such as C and B) do not add a timestamp or nonce to the message. Replay attacks can be detected by the unique combination of route_list, RREQ_ID, T_D and N_D without the need for additional message elements.

Upon receiving such a message, B validates at least the certificate and signature of C (i.e. the immediate sender). It could validate all the signatures, but this is better left to the original sender. As the RREQ was broadcast B will not know which other node to expect the RREP. All it can realistically do is assure itself that the immediate was message was indeed sent by the claimed sender. It then checks for replays, as above and generates the following message for the previous hop (in this case the original sender, S).

(5) B → S: RREP-U_D, RREQ_ID, L_D, T_D, N_D, route_list, Cert_D, Sig_D, Cert_C, Sig_C, Cert_B, Sig_B

where

\[ \text{Sig}_B: \{ \text{RREQ\_ID, L_D, T_D, N_D, route\_list} \} S_{K\cdot B} \]

\[ \text{route\_list: D, C, B, S} \]

Message (5) is essentially identical to message (4), with the additional route information. S may now validate the included certificates, validate the signatures using the public keys in those certificates, check for replay and ensure the lifetime has not expired. It has a route list, guaranteed by the signatures of each node along the route, and can decide whether the route consists of trustworthy nodes.

If the original sender decides that it will accept the route it records the route as an active route to the destination, as per standard AODV. It should also store the route list, signatures and certificates, to enable it to act as an intermediate node with an active route to the destination in the event that it receives from another node a subsequent RREQ for a route to the destination.

\(^3\) If nodes do not wish, or are unable, to store this information, they simply re-broadcast RREQs.
10.6 Secure Route Maintenance

A node attempting to forward a packet to the next hop along the source route returns a route error (RERR) to the original sender if it is unable to deliver the packet to the next hop after a limited number of transmission attempts. Ideally unauthorised nodes should be prevented from sending RERR messages. In practice this is impossible so we require that a sender sign such a message. This will allows the source of each RERR message to be traced and simple forgeries discarded. Each node on the path must forward this message towards the source.

Reverting back to the previous example, if during packet transmission the node $C$ is unable to reach $D$, $C$ must send a RERR message to the source $S$. The extended RERR message is as follows:

$$C \rightarrow B: \text{RRER-U}_C, C, T_C, N_C, \text{Cert}_C, \text{Sig}_C$$

where

$$\text{Sig}_C: \{\text{RRER-U}_C, C, T_C, N_C\} S_{K-C}$$

The objective of message (6) is for $C$ to securely indicate that it is unable to deliver the packet to the destination. The addition of a signature to the message allows any receiving node (specifically the original node) that the RERR message did indeed originate at $C$. The timestamp, $T_C$, and nonce, $N_C$, help prevent replay attacks. Upon receipt of this message the sender can check the certificate and signature. If both are successfully checked it is then guaranteed that $C$ did actually send such a message and that the message has not been altered in transit. This scheme thus provides for integrity protection as well as non-repudiation. The contents of the standard RERR message allow the original sender to identify the route that has failed.

10.7 Remarks

The protocol modifications defined above fulfils our stated goals as follows.

It should allow identification of the sender of a message. All messages now included a signature, transmitted with the rest of the message. The certificate of the signer is also included, allowing nodes to verify the certificate. By validating the signature a receiver is assured that the message was originally sent by the signing node and no other. As the amended message structures include timestamps and nonces and the signature is computed over these elements, our proposal allows replay attacks to be detected. Also since the signature can be validated using the private key of the sender of a message it helps to protect against repudiation attacks.

It should be possible to detect if a message has been altered in transit. Verifying the signature will assure a recipient of a message not only of the identity of the sender but also that the message received is the same as the message sent.

It should be possible for the initiator of a RREQ to discover all the nodes that make up a path when it receives a RREP. As the RREP is conveyed back to the original sender the route list to the sender is constructed. Each node signs its contribution to the route list. The original sender may verify these signatures from the certificates also included in the RREP. The route list components include the identity of the current node and
the identity of the next node back in the bath to prevent additions to or deletions from the route list.

Our proposal also helps prevent flooding of the network with unnecessary request messages. The original sender signs an RREQ message. This allows senders to be identified and “nuisance” nodes that send multiple RREQ messages can be detected. Replays of RREQ messages can be detected by examining the timestamp, nonce and RREQ_ID fields.

As the nonce and timestamp are placed under a signature it is not feasible for an attacker to tamper with this information. Also since timestamp and remaining lifetime of messages are recorded at each hop, it makes it difficult for the attacker to indulge in rushing attacks. The source route characteristic of the protocol itself prevents routing loops from happening.

Note that the route list is built on the way back to the originator (in the RREP messages), not on the way out (in the RREQ messages). Each node checks that it is in the list and adds to the list the node to which it will next send the RREP. This prevents the presence of a node in the list being dropped by any node closer to the originator, as the signatures would then compute correctly. Building the route list on the way out would require the partial route list to be carried in the broadcast RREQ messages. This would be excessive in bandwidth requirements, especially in those parts of the broadcast tree that do not lead to a RREP. In contrast, the RREP messages are unicast, limiting the bandwidth required to carry the route list. The drawback of this approach is that only the originator, not the destination, can check the route list. The source could send the route list to the destination, but this would extra messages to the AODV exchange (something we attempted to avoid).

As we do not aim to provide anonymous routing, passive attackers can eavesdrop on routing messages especially route requests. One attack our extensions to not protect against is the deliberate dropping by intermediate nodes of RREQ/RREP/RERR messages. Protection against this denial of service attack is left to future work.

**10.8 Security related extensions to AODV Messages**

The protocol extensions described in sections 6 and 7 above require additions to the existing message formats of AODV. In this section we describe the extensions to the RREQ, RREP and RERR messages.

Figure 16 depicts the extended format of the RREQ message. The extensions are divided into two parts – those added by the original sender and those added by an intermediate node re-broadcasting the message. The length field gives the length, in bytes, of the extensions. While created by the originating node it will be updated at each intermediate node. The hash function and sign algorithm fields specify the algorithms used to compute the signature. The lifetime and timestamp fields contain the relevant values. The certificate length field specifies the length of the sender’s certificate and this is followed by the actual certificate and the original sender’s signature. Both certificate and signature would probably be longer than the four bytes shown (a receiver being able to determine their length from the certificate length and
sign algorithm fields). The remaining fields are not included by the original sender, but added and updated by intermediate nodes.

This can be detected from the hop count and length fields in the message. This section begins with a field for the intermediate node’s IP address and followed by other fields similar to those for the original sender.

![Figure 16: Extensions to RREQ Message](image)

Figure 17 depicts the extended format of the RREP message. The extensions consist of three parts. The first part consists of the RREQ_ID, lifetime, timestamp and nonce fields added by D. These fields allow a node receiving this message to detect a replay. Second is the route list and associated length field (number of nodes). This is commenced by D, by incorporating its own address, and the address of the next node back along the route. Note that (apart from D itself) a node’s address is added by the previous node in the route list, not the node itself. As the address of this next node is included in the signature, a subsequent node (including the node itself) cannot delete it from the route list in an undetectable fashion.

The final part of the extensions is the certificates and signatures of the nodes contributing to the route list. The fields hash function and sign algorithm specify the methods used to construct each signature.

Figure 18 depicts the extensions to the RRER message. These are simple, but conform to the pattern established for the other messages. The node detecting a route
error adds its own IP address to the message. The receiving node (originator of the route) can use this in deciding whether to trust the message. *Timestamp* and *nonce* fields are used to help prevent replay attacks. The sender of the RERR messages adds a signature, for integrity and authentication and a certificate to allow the signature to be checked.

![Figure 17: Extensions to RREP Message](image)

![Figure 18: Extensions to RERR Message](image)
10.9 Comparison With Other Approaches

There are several approaches to providing security for routing protocols for ad-hoc networks. Here we shall focus on only those solutions that aim to provide security for AODV.

One approach was proposed in [Venkatraman+ 01]. This approach describes a scheme for inter-router authentication as well as a scheme to handle replay problems that could prevail using existing schemes. The scheme has been incorporated into the AODV routing protocol to minimise security threats. Just as in our approach, this model offers the integrity of the routing requests. The hash of the message is computed and then encrypted with the sender’s private key for authentication. When the neighbours receive the route request, the validity of the node is verified using the sender’s public key. This scheme, like our approach, offers strong authentication for replay messages. In our scheme the number of control messages to achieve the same is kept to minimum (one). Since a challenge-response mechanism is used in [Venktaraman+ 01], a three-way communication occurs between every pair of intermediate nodes increasing the overheads considerably. This also implies that the basic functionality of the protocol has been modified to introduce security, which we have avoided. We have also added security to the RERR messages, an aspect ignored in [Venkatraman+].

In [Zapata+01], a security extension to the AODV protocol is proposed. Similarly to our approach, this scheme proposes security extensions in a seamless manner without impinging upon the actual functionality of the protocol. The goal of this scheme is that if a node plans to build an attack by not behaving according to the AODV protocol its damage will be restricted as it can only selectively not reply to certain routing messages and can only lie about information about itself. Unlike our scheme, the lifetime field is not very strongly authenticated as intermediate nodes. As intermediate nodes are allowed to reply to RREQs, they can lie about the lifetime. In our scheme the intermediate node must use the lifetime supplied, and signed, by the ultimate destination. This approach also only identifies, and has signatures from the original source of a message. Our scheme also identifies the immediate re-sender of RREQ and RREQ messages, thus allowing better protection against replay and denial of service attacks. The scheme in [Zapata+01] is also weaker against replay attacks due to the lack of timestamps and nonces in the message extensions. This will make it harder for nodes to detect replays in that scheme. Finally, we provide a route list in the RREP while [Zapata+01] does not. Without this route list it is impossible for a sender to make any judgements about a returned route.

10.10 Summary

Routing protocols that have been proposed for use in ad-hoc networks are highly susceptible to a number of different attacks. These could not only degrade the performance of the network but could even bring the communication to a complete halt. In this thesis, we have proposed security focus extensions to the ad-hoc on-demand distance vector (AODV) routing protocol. We identified several threats specific to the AODV environment and corresponding services that counteract these threats. We then proposed a security framework by applying security to the existing control messages in AODV. These protocols meet our previously identified design
goals. We then discussed extensions to existing AODV messages to facilitate these services. Finally, we compared our solution with other prominent approaches that aim to secure AODV. Our approach seamlessly integrates into the existing AODV framework without impinging upon the protocol functionality.
Chapter 11

Secure QoS Layer for NTDR Ad-hoc Networks

11.1 Introduction

Quality of Service (QoS) is the capability of a network to provide better service to selected network traffic over various underlying technologies such as Frame Relay, ATM, IP and routed networks. In other words, it is that feature of the network by which it can differentiate between different classes of traffic and treat them differently. The provision of QoS features is increasingly becoming a critical issue in today’s networks as there is a growing need for networks to support multiple kinds of traffic due to growing user demands and the presence of multimedia applications. But these different classes of traffic must be supported over single network links. Different kinds of traffic demand different treatments from the network. Therefore the challenge is to design a framework where certain classes of traffic can get preferential treatment over others and yet at the same time all these classes of traffic can share the bandwidth efficiently.

Several Schemes propose QoS based solutions for wired networks. However none of these solutions scale well for wireless ad-hoc networks due to their peculiar nature of having no fixed wireless infrastructure and dynamic topology. These problems are further aggravated due to factors such as poor quality of wireless links, resource limitations and frequent and rapid mobility of hosts. These problems must be addressed for widespread deployment of ad-hoc networks. The aim of this chapter is to address QoS requirements for mobile ad-hoc networks.

We begin this chapter by introducing the concept of QoS and protocols that support QoS in section 11.2. In section 11.3, we discuss some of the challenges that one faces when addressing QoS issues for wireless ad-hoc networks. We also review some current research done in this area. In section 11.4, we introduce our framework for providing QoS for mobile ad-hoc networks. Our proposal aims to provide QoS for wireless NTDR networks. Since security is an important aspect of any QoS framework, our proposal includes security as part of the QoS solution. In particular, we propose design and placement of a Secure QoS (S-QoS) layer for wireless NTDR networks. As we intend to provide QoS related services on an end-to-end basis we focus exclusively at the IP layer of the protocol stack. This does not in any way undermine the need for appropriate QoS related services at the link layer. In fact, for any QoS framework to operate efficiently at the IP layer and above requires a robust QoS Medium Access Control (MAC) protocol. Therefore QoS routing at the IP level as well as QoS services at MAC layer must cooperate to provide a complete QoS solution. In section 11.5 we discuss the design of a Management Information Base (MIB) that is necessary to support the proposed layer. Finally, in section 11.6 we provide our concluding remarks.

11.2 Introduction to QoS

Quality of Service (henceforth QoS) is described in [ITU E.800 94] as “the collective effect of service performance, which determines the degree of satisfaction of a user of the service”. While it can be argued that today more and more bandwidth is available
at increasingly low price and therefore QoS is not really an issue, the bursty nature of data makes it critical for some QoS control to be present in the network. The bursty nature of data traffic leads to congestion for short periods in some nodes and segments in the network, irrespective of the fact that a large capacity is available. The other problem is that the existing routing protocols are not aware of load levels in the network; therefore congestion will build up on some paths while others have bandwidth to spare. When considering wireless networks, it must be noted that there is also a speed mismatch at the wired and wireless network interface. With wired networks capable of supporting increasingly high speeds, it is likely to congest the wireless link. Finally, bandwidth alone does not ensure low and predictable delay, as even with huge bandwidth, there is still the danger that real-time applications will get stuck behind large file transfers. Therefore, QoS mechanisms are important because they enable networks to deliver defined levels of service with the existing network infrastructure. There are three generic methods that cater to traffic in a network: Best Effort framework, Integrated Services framework and Differentiated Services framework.

11.2.1 Best Effort Framework

In this framework, an application sends data whenever it wishes, as much as it wishes and without requiring anyone's permission. The network elements will attempt to deliver the packet without any bounds on QoS parameters such as delay, latency, and jitter but if problems do occur, the traffic is discarded. In other words, with best effort service, timing between the packets is not guaranteed to be preserved, packets are not guaranteed to be received in the order in which they were sent, nor is the eventual delivery of transmitted packets guaranteed [Kurose+01]. So the onus is on the end systems to make sure that the packet goes through. The current day Internet packet delivery system is an example of this service. Due to the lack of any special effort to deliver packets in a reliable manner with no guarantees, it is extremely challenging problem to develop successful multimedia networking applications for the Internet.

11.2.2 Integrated Services (IntServ) Framework

This is a set of standards set down by the IETF to provide individualised QoS guarantees to individual application sessions. Therefore it becomes mandatory for the applications to know the characteristics of their traffic before hand and signal the intermediate network elements to reserve certain resources to meet its traffic properties. The network elements such as the router are also required to know how much of their resources such as buffer and link bandwidth are already reserved for ongoing sessions. According to the availability of resources, the network either reserves the resources and sends back a positive acknowledgment, or answers in the negative. This part of the standard is called Admissions Control. This decision will usually depend on the traffic specification, the requested type of service, and the existing resource commitments already made by the router to ongoing sessions [Kurose+01]. It is often decided by the policy decisions of the router/switch. In the absence of any admissions control, it would mean granting all available resources to all classes of traffic, which is what best effort networks offer. If the network agrees to support an application request, the application can send the data but must adhere to the traffic properties it has negotiated with the network. If however, the application sends traffic that is out-of-profile, then the data is given best effort service, which may
cause the packets being dropped altogether. This signalling protocol is called Resource ReSerVation Protocol (RSVP). The Intserv framework defines two major classes of service: guaranteed service and controlled load service.

**Guaranteed QoS**

According to this specification [RFC 2212], there are tight bounds on the queuing delays that a packet will experience in a router. In principle this service is realised using a leaky bucket algorithm. Traffic is specified using a leaky bucket characterisation. Since the user’s requested transmission rate is known beforehand, using this detail with leaky bucket characterisation, it is possible to bound the maximum queuing delay at the router [Kurose+ 01].

**Controlled Load Service**

In this type of service [RFC 2211], an application will receive a QoS service closely approximating the QoS that same flow would receive from an unloaded network element. In other words, an application may assume that a very high percentage of its packets will successfully pass through the router without being dropped and will experience a queuing delay in the router that is close to zero [Kurose+ 01]. This service targets real-time multimedia applications that have been developed for today’s Internet. As previously mentioned, these applications perform well when the network is unloaded, but rapidly degrade in performance as the network becomes loaded.

**RSVP: Intserv Signalling Protocol**

As mentioned above, for a network to provide QoS guarantees, there must be a signalling protocol that allows applications running in hosts to reserve resources in the Internet. This is made possible by using RSVP [RFC 1633] as a signalling protocol. A data flow in RSVP is a sequence of messages that have the same source, destination and the same QoS. In RSVP resources are reserved for unidirectional data or "simplex" flows going from sender to receiver. The sender is upstream and the receiver downstream.

The protocol begins when a host application wanting to establish a session with some target sends a special data packet called the **PATH** message. This message is destined for the intended receiver. This packet has the characteristics of the traffic the sender is going to send in the forthcoming session. At each router, the **PATH** message attempts to make a resource reservation for the stream. The router and intermediate forwarding devices install a path state with the help of this **PATH** message and become aware of their adjacent RSVP aware devices. Thus a path from the source to the destination is pinned down. If the path cannot be installed then a **PATH** Error message is sent upstream to the sender who generated the **PATH** message.

After the receiver gets the **PATH** message, it issues a **RESV** message. The objective of this message is to furnish the details of the actual QoS characteristics expected by the receiver. Different receivers may specify different QoS features for the same multicast flow. **RESV** exactly traces back the path taken by the **PATH** message. Thus each device along the path gets to know the actual QoS characteristics of the flow requested by the receiver and each decide independently how much of the demand it
should satisfy or refuse altogether. If it refuses then a RESV Error message is issued downstream to the receiver who generated it in the first place.

It is important to note that RSVP is merely a protocol that allows applications to reserve the necessary link bandwidth but does not specify how the network provides the reserved bandwidth to the data flows [Kurose+ 01]. Also, RSVP is not a routing protocol, as it does not determine the links in which the reservations are made. Instead it depends on the underlying routing protocol (unicast or multicast) to determine the routes for the flows.

**Observations**

The RSVP based Intserv approach has some problems. The per flow based reservation processing prescribed in this approach represents a considerable overhead in large networks. With recent measurements [Thomson+ 97] indicating that even for an OC-3 speed link, approximately 256,000 source-destination pairs might be seen in one minute in a backbone router. The presence of an Intserv will increase introduce large overheads in such situations. Also, the Intserv framework caters to only a small number of pre-specified service models. This particular set of service classes does not allow for more qualitative or relative definitions of service distinctions [Kurose+ 01]. These issues have initiated development of a new framework called the Differentiated Services Framework, described in the next section.

**11.2.3 Differentiated Services (DiffServ) Framework**

According to this model, network traffic is classified and conditioned at the entry to a network and assigned to different traffic classes known as behaviour aggregates [RFC 2475]. These behavioural aggregates are a collection of packets with common characteristics as far as how they are identified and treated by the network. Each such aggregate is assigned a single codepoint known as the Diffserv (DS) codepoint. In other words, the DS codepoint is the mark that a packet receives that helps in identifying the class of traffic to which it belongs. In the core of the network packets are forwarded as per the per hop behaviours associated with the codepoints.

After the packets have been classified at the boundary of the network, they are forwarded through the network based on the DS codepoint. The forwarding is performed on a per hop basis; that is, the DS node alone decides how the forwarding is to be carried out. This concept is called the Per Hop Behaviour (PHB). The PHB is the observable forwarding behaviour of a DS node. In order to provide a controlled environment and prevent congestion, the traffic conditioning functions must enforce rules on the influx of traffic into and out of the DS domain. These rules are known collectively as the Traffic Conditioning Agreement (TCA).

**DiffServe Architecture**

DiffServ divides the network into domains. A domain can be defined as a set of nodes, which support a common resource provisioning and PHB policy. It has a well-defined boundary and there are two types of nodes associated with a DS domain - Egress nodes and Interior nodes. Egress nodes connect the DS domain to other domains. Interior nodes are connected to other interior nodes or boundary nodes - but
they must be within the same DS domain. The task of classifying the packets by marking them is done at the boundary nodes. They also enforce the Traffic Conditioning Agreements (TCA) between one DS domain and the other domain it connects to. Interior nodes map the DS codepoints of each packet into the set of PHB’s and impart appropriate forwarding behaviour. Any non-DS compliant node inside a DS domain results in unpredictable performance and a loss of end-to-end QoS. A DS domain is generally made up of a organization's intranet or an ISP - i.e. networks controlled by a single entity. The DiffServ architecture can be extended across domains by the provision of Service Level Agreements (SLA) between them. A SLA specifies rules such as for traffic remarking, actions to be taken for out-of-profile traffic, etc. The TCA between domains are decided out of this SLA.

Observations

The DiffServ approach minimises the signalling overheads encountered in the Intserv approach by aggregation and per-hop behaviour features. Flows are classified by predetermined rules so that they can fit into a limited set of class flows. This eases congestion from the backbone. The edge routers use the 8-bit Type of Service (ToS) field, called the DS field in DiffServ terminology, to mark the packet for preferential treatment by the core transit routers. 6 bits of it are currently used and 2 are reserved for future use. Also only the edge routers need to maintain per-flow states and perform the shaping and the policing. Therefore computation intensive traffic shaping and policing strategies reside only at the edge routers and not at the core. Therefore once the packets get to the core they can be routed very fast incurring minimum computational delay at any router/switch.

11.3 Quality of Service Issues in Mobile Ad-hoc Networks

Two related issues significantly affect quality of service in an ad-hoc network environment:

- The continuously varying and unpredictable QoS that a mobile node experiences
- The overhead associated with the QoS negotiation and resource allocation in the network.

As the mobile node moves the available bandwidth and the bit error rates will change more or less continuously. These conditions must be overcome for the continuous operation of the network.

Ad-hoc networks have no fixed network infrastructure or administrative support. Therefore there is no dedicated entity to manage the network or its resources or for that matter enforce policy for network management. This implies that the necessary control and admission policies for implementing a quality of service framework must be done exclusively by interaction between the participating nodes. These nodes must participate together to allocate necessary resources that are needed to support a QoS session on an end-to-end basis. In such networks, the mobility of nodes may be very high resulting in frequent changes in topology. If resources were reserved prior to topological change then the issue arises whether a new route is available or not to the intended destination node and in case if it is available then are there enough resources available in the new route to meet previously negotiated resource requirements.
Unlike a traditional wired network, in an ad-hoc network environment each participating node functions as a router as well. End-to-end QoS guarantee depends upon the availability of resources at each such node along the path to the destination. A node break down or its going into the doze mode, may result in the path being unavailable causing the network or source to re-establish the path causing undue delay in the delivery of packets belonging to a session. Also frequent changes in topology may result in re-negotiating the QoS service parameters and reallocating resources that can considerably increase delay jitter experienced by the users as this increases the computational complexity of the route selection criteria.

Mobile ad-hoc networks run on top of wireless link layer protocols that are contention-based. The larger the number of nodes in the network, the greater is the delay in accessing the channel for transmitting a packet. Also the choice of the link layer influences the QOS framework at the network layer. The link layer must provide support for multiple access techniques with suitable collision avoidance mechanisms and must also mitigate the ill effects of the hidden terminal problems as these issues can significantly degrade the overall end-to-end QoS. There can be variations in link quality due to blind spots under bridges, behind buildings or hills or due to atmospheric conditions such as rain or lightning. These effects require more sophisticated dynamic QoS management than wired and traditional wireless networks.

An increase in the size of the mobile ad-hoc network increases in turn the complexities of the computational load, there is a difficulty in the propagation of network updates within the given time bounds. This is a factor that determines whether the network is combinatorially stable or not [Chakrabarti+ 01]. An ad-hoc network is combinatorially stable if and only if the topology changes occur sufficiently slowly to allow successful operation of all topology updates as necessary. Otherwise the network topology information at each node becomes inconsistent resulting in the degradation of communication.

Different service types have different objectives for delay, bandwidth and packet loss. Hence not all packet exchanges will be treated with equal priority in a QoS enabled network. For instance the priority assigned to the control traffic flow will be higher than that for regular data packet flow. Also for data packets themselves, different flows will have different priorities. There are issues of handling user data with multiple priorities. The network must first authenticate a user’s request for QoS. This authentication process must not be so complex and cumbersome as to degrade the operational performance of the network [Chakrabarti+ 01]. Having processed this request the network attempts to find an appropriate route to the destination that satisfies the user’s quality of service requirements. In heavy traffic situations guaranteeing QoS for lesser priority traffic can be extremely difficult. The development of QoS routing policies, algorithms, and protocols for handling user data with multiple priorities is also an open research area [Chakrabarti+ 01].

Finally security is a critical component of any QoS framework. The objective of a security policy for an ad-hoc network is to provide secrecy to user data, authentication of participating nodes as well as integrity of routing protocol information. A node may also launch malicious attacks by flooding the network with too many invalid requests. We have dealt with issues relating to security in previous chapters. The point
that we are trying to make here is that security is tightly coupled with QoS and any framework for the provision of QoS automatically must include security.

11.4 Secure QoS Layer

11.4.1 Introduction

We define a Secure Quality of Service (S-QoS) layer for NTDR Mobile ad-hoc networks. The purpose of the S-QoS layer is to provide various quality of service and security services in the network layer through the use of cryptographic mechanisms. It provides support for these services via label-based mechanisms as in MPLS. In principle, our S-QoS layer can be seen as an extension to the shim header proposed for MPLS based communications. This protocol requires the services of QoS Management of Information Base (S-QoS_MIB) as an external service that governs its operation and maintains it. It also requires the presence of an underlying routing protocol for negotiation of QoS related attributes and efficient distribution of labels required to operate and maintain such networks. For reasons explained below, we believe that on-demand routing schemes provide a more convenient way to negotiate labels for user data based traffic and table based schemes are appropriate for negotiating state information between clusters. In principle, the QoS facility is provided by using a combination of Intserv and Diffserv schemes described above to achieve scalability and to reduce the overheads.

We envisage a scenario where several clusters under different clusterheads form a domain. Each clusterhead acts as an ingress node for nodes in its cluster. The following classes of traffic have been predefined for this environment. The constant bit rate (CBR) traffic with high priority followed by VBR non-real time traffic and then VBR real time traffic. The behavioural aggregates map on to these predefined traffic classes. A label is associated with a traffic flow that indicates to which aggregate this flow belongs. We take AODV as an example to establish a QoS state at each node for a traffic flow. AODV messages serve the purpose of signalling messages to establish a QoS state at each node while the S-QoS layer is responsible for data packets of traffic flows belonging to a particular traffic class. We therefore divide the provision of quality of service into two phases: the signalling phase using AODV and data transfer phase using S-QoS. In order to support this framework, AODV messages must be extended to accommodate QoS parameters for negotiation. The following sequence of steps takes place during the signalling phase.

1. A source node S in a cluster generates a request message for its clusterhead to establish a communication session with a destination node D. The request message amongst other things contain the following QoS related parameters:
   - The destination node id (D).
   - Bandwidth requirement B.
   - Traffic description parameters.
   - Security specifications.

2. Since all the traffic must flow through the clusterhead (a requirement in the NTDR network), the clusterhead must first determine if it can accommodate this traffic flow request (admission control function). If yes, then the clusterhead
places this requested traffic flow into a particular traffic class by assigning it an appropriate label. Thereafter the clusterhead refers to its routing table to locate the destination node D (route check to destination). If the destination is not found it generates the AODV RREQ message to all the neighbouring clusterheads (flooding). The RREQ message is extended to accommodate the above-mentioned QoS parameters along with the label class.

3. Each node along the path performs the admission control check based on the label assigned as well as the route check and then if need be it resorts to flooding. If a node is unable to satisfy the QoS requirements, it must discard this packet. This process is repeated till the RREQ message reaches the intended destination D. On the other hand, if a node is able to locate a route to the destination then instead of flooding the packet, this node directs the request along the path indicated by the routing table.

4. The acknowledgement from D (RRESP) message traverses the same path back towards the source confirming the reservation of resources on the way.

11.4.2 Placement of S-QoS Layer in the Protocol Stack

In this section we consider the various options available for the placement of S-QoS services within the ad-hoc network environment. The two related issues to consider are as follows:

- Should the mechanisms, which provide these services, be kept together or be distributed through the protocol stack?
- Where precisely should these mechanisms be placed in the protocol stack?

While it is possible to distribute these mechanisms throughout the protocol stack we believe that it is preferable to collocate all the mechanisms together at one point as far as possible, as this will minimise the modifications to the protocol stack and allow mechanisms to share cryptographic components. Furthermore, it will simplify the implementation of any synchronisation schemes for the cryptographic mechanisms.

In [Varadharajan 97], the above-mentioned issues in the design of high-speed networks are addressed. This work reviews the existing security protocols for TCP/IP and OSI networks and assesses their suitability for providing security in broadband networks. These developed arguments are then applied to design security services for connection-oriented Frame Relay networks. We base our investigations on the arguments presented in this work to support and formulate our design goals.

There are various options available for the placement of S-QoS related services in the ad-hoc network environment. The Mobile Ad-hoc Network protocol stack essentially consists of TCP/IP protocol suite running over wireless link layer protocols such as 802.11 or Bluetooth. Theoretically, it is possible to place these mechanisms at any layer of this protocol stack, each option having its own advantages and disadvantages. We now consider the following options:

- **Option One:** S-QoS Functions at the Link Layer.
Option Two: S-QoS Functions above the Network (IP) layer (This includes all the options available for the placement of the layer from the transport to the application).

Option Three: S-QoS Functions between the IP and the link layer.

Option One: S-QoS Functions at the Link Layer

Any QoS related infrastructure assumes the presence of an underlying MAC protocol to provide services that cater to medium contention, reliable unicast communication and resource reservation. Several MAC protocols have been proposed such as [Fullmer+ 95, Karn 90, Bhargavan+ 94 and Talucci+ 97]. Unfortunately, their design goals are usually to solve medium contention and hidden/exposed terminal problems and to improve throughput [Wu+ 01]. They do not provide QoS features to support real time traffic. The GAMA/PR protocol and the Black Burst contention mechanism [Sobrinho+ 99] aim to provide QoS guarantees to real time traffic in a distributed wireless environment. However their use is restricted to wireless LAN environment wherein every host can sense each other’s transmission. Recently there have been proposals that extend the IEEE 802.11 protocol [IEEE Std. 802.11 99] to provide QoS related features. However, when considering provision of QoS on an end-to-end basis there are some issues that need to be taken into account. QoS for wireless LAN applications and users has been limited to the radio frequency. Also there can be situations where several clusters need to communicate with each implementing a different link layer protocol. Each link layer will then have a different type of QoS technology that can be applied further complicating the matters. Also the placement of security services at link layer introduces additional overheads due to hop-by-hop encryption and decryption operations.

Option Two: S-QoS Functions above the Network (IP) layer

A possible choice will be to place the S-QoS related mechanisms at the transport layer. Reliability and flow control on an end-to-end basis are inherently the services of the transport layer. These features can be exploited by S-QoS layer mechanisms to provide an elaborate and complete end-to-end QoS service. Furthermore, from the security perspective connection oriented confidentiality and integrity service can be provided conveniently at this layer. Since QoS is meaningful only for a flow of packets between the source and destination, and thus depends on the notion of a logical association, or logical connection between them for the duration of flow [Chakrabarti+ 01], it is intuitively clear that use of connection oriented security services makes sense in such environments. It may also be useful to integrate QoS mechanisms at the application layer. Since different applications require different types of QoS, any such solution can get to be highly application specific. Also any solution that prescribes the provision of QoS above the network layer is independent of the underlying ad-hoc network technology.

Option Three: S-QoS Functions between the Network (IP) Layer and the Link Layer

Since IP is the predominant networking protocol, there are several solutions and standards such as Intserv, Diffserv and MPLS that support IP QoS framework. Also since the link layer specifications may change, IP remains the same thus integrating
various underlying protocols to provide consistent implementation of QoS services across the network. However, there are some issues that need to be considered. If the layer resides between the IP and the link layers then it essentially operates on IP datagrams. IP is a connectionless datagram service, that is it does not provide sequencing and reliability features on an end-to-end basis. It only operates on a per protocol data unit basis. It may be argued that from the security perspective, provision of security services leads to a major simplification of the security protocol as the needed reliability can be provided by the transport protocol that will operate above the security protocol. However, this has serious implications on the provision of QoS since QoS is meaningful not on a per packet basis, rather on a packet flow basis. Therefore, mechanisms must be in place to identify a packet flow at the IP level to provide connection-oriented services. If such a facility is available, then IP seems to be the most obvious choice to accommodate S-QoS related functions.

One obvious choice is to use IPSec [RFC 2401]. AS IPSec provides encryption at the network layer, applications do not need any modifications in order to take advantage of the capability. Encrypted IP packets look just like ordinary IP packets. As a result, they can traverse the Internet, extranet, and intranets just as easily and transparently as ordinary IP packets once the end station has been equipped with IPSec technology. However there are some serious issues associated with the usage of the IPSec Protocol. IPSec does not address many of the MANET specific security issues such as denial of service attack on the routing infrastructure or selfish nodes not participating in the routing function. Furthermore protocols like IKE are too complex to be implemented in small devices and normal PKIs are also not suitable in this context. So there is probably a need for either a different approach or at least augment IPSec significantly to really provide security for MANETs. Furthermore IPSec was designed to facilitate security related functions and not QoS. Extending IPSec to provide the necessary QoS features will make the protocol far too complex to be implemented in the ad-hoc environment.

11.4.3 S-QoS Layer Functions

Provision and management of QoS services includes various aspects that pertain to the nature of perceived quality. In this section, we shall identify critical QoS related functions that need to be supported by a QoS layer. Identification of these parameters directly impacts the design principles governing the S-QoS layer. Before we begin to identify these parameters, we need to identify different traffic types in the network. In our model we identify two types of data channels: Constant Bit Rate (CBR) and Variable Bit Rate (VBR). CBR service is used for applications and connections that require a fixed and consistent quality of bandwidth. Bandwidth and resources on a CBR connection is established and fixed at peak information rate (PIR) for the duration of the connection where PIR is a traffic parameter that characterises the source and gives the maximum rate at which packets can be transmitted. VBR traffic is similar to CBR, but the amount of data flowing at any instant can vary somewhat. The flow is characterised by an average bit rate and peak rate implying that usually the traffic will be at average bit rate but occasionally it might be at peak rate. This allows for occasional bursts of data to be carried without loss as long as the average rate is not exceeded over a long span of time. VBR traffic can be further divided into real-time and non-real-Real-time. Real-time VBR traffic (VBRrt) is sensitive to delays while non-real time VBR traffic (VBRnrt) is more tolerant of delay.
We shall now focus on functions that S-QoS in conjunction with the S-QoS-MIB layer must support to provide the desired level of QoS. In conjunction with this MIB, the S-QoS layer aims to provide the following services:

1. **Classification of Traffic**: The S-QoS layer must be able to identify a traffic stream belonging to a particular class (CBR/VBR/ABR) and be able to provide the necessary services associated with that particular class of traffic. When a node requests for a QoS service, then the presence of ABR and VBR parameters in the request makes it evident that the node desires a VBR service. In the absence of ABR, the request is for a CBR service. In regular data flow, the packets must have an identifier to mark it either as CBR type or ABR type.

2. **Reliable or Unreliable Traffic Type**: This specifies whether data transferred may be lost (real-time data) or not (non-real time data). The node in its request will specify this requirement using a parameter called *Loss_Ratio*. This *Loss_Ratio* is a QoS parameter that gives the ratio of the lost cells to the total number of transmitted cells. If this parameter is set to zero, then the request is for a reliable session (non-real time), otherwise the session is unreliable (real-time). In order to achieve the reliability of data flows, appropriate sequence numbers must be associated with the data packets belonging to that flow. This reliability feature can be further enhanced by the use integrity verification mechanisms explained below.

3. **Bandwidth of a Communication Session**: This specifies the bandwidth requirements of a communication session. This can be determined during the request by examining the traffic descriptor parameters. In other words, if the type of channel is CBR, then the bandwidth desired is the same as *PIR* value. If the type of request is for a VBR session, then the bandwidth is the same as the ABR value. Also appropriate packet scheduling mechanisms must be in place to allocate and manage the bandwidth requirements of a session.

4. **Prioritising Traffic Flows**: There must be a mechanism in place to prioritise the traffic flows in the network. CBR traffic flows are given a higher priority over VBR traffic flows. With the VBR traffic, VBRnrt is to be given a higher priority over VBRrt. The highest priority class must be reserved for control messages. The packet header of a packet belonging to a particular flow type must carry the priority associated with this packet. The header must also include appropriate field that differentiates a control packet from a data packet.

5. **Aggregating traffic Flows**: There must be mechanisms in place by means of which several individual traffic flows having similar characteristics can be processed together without having the need to incur the overheads involved in processing each flow individually. This can be achieved by allocating appropriate identifiers called labels that bind traffic flows with similar characteristics. The label field of the packet is then examined to determine the treatment it deserves.

6. **Security Services for Traffic Flows**: There must be provision for security services such as privacy of user-to-user communication sessions, integrity of control messages, authentication between communicating parties as well as
intermediate nodes and some coarse-grained access control services using the appropriate key service.

Having identified the functional requirements of the S-QoS framework, we now proceed to examine the S-QoS layer design.

11.4.4 S-QoS Layer Design

The S-QoS layer resides below the IP layer and above the data link layer. It can be defined as a quality of service extension to the IP protocol using a label based technique as in MPLS. In principle, it extends the functionality of the MPLS shim header to facilitate both QoS as well as security services. The S-QoS layer depends upon some control mechanism for the exchange of QoS attributes. Such a control protocol could be an extension of label distribution protocol specified in [RFC 3036]. The specification of such a protocol is beyond the scope of this thesis. However, we suggest extensions to the Ad-hoc On Demand Routing Protocol (AODV) to provide some basic QoS negotiation as well as label exchange features. The S-QoS packet consists of a clear header, a protected header, protected data (Payload) and integrity check. The packet format is shown in Figure 19. S-QoS layer in conjunction with S-QoS_MIB facilitates the service functions that were mentioned above.

![Figure 19: S-QoS Protocol Data Unit](image)

Clear Header

Figure 20 shows the format of the clear header. All fields are required and must appear in the order shown.

![Figure 20: S-QoS Protocol Data Unit Clear Header](image)
**Label:** Identifies the QoS associated with a traffic type.

- **Label Id:** It is associated with a traffic aggregate type. It indicates the QoS to be assigned to the traffic aggregate type.
- **Type:** It indicates the traffic type (CBR, VBR).
- **Priority:** It indicates the priority associated with the traffic type.

**Length:** This is a length indicator field that contains the length of the clear header in octets. The value includes the length field.

**SE:** This field contains a value that indicates whether the security option is enabled for this PDU type or not.

**SAI:** This is the Security Association Identifier Field. This field must be present if security in the SE field is set as enabled. This field identifies the security attributes (such as keys for confidentiality and integrity) associated with this PDU. The corresponding MIB entry is indexed with the SAI field.

**Seq. No.:** This field identifies the sequence number associated with the packet. It is used to provide reliability.

**N-List:** This field indicates the QoS enabled path between the source and destination that this PDU must traverse. This path was determined during the route establishment phase using AODV.

**Protected Header**

Figure 21 shows the format of the clear header. All fields are required and must appear in the order shown.

<table>
<thead>
<tr>
<th>Length</th>
<th>Label</th>
<th>SAI</th>
<th>Seq. No.</th>
<th>N-List</th>
</tr>
</thead>
</table>

**Figure 21: S-QoS Protocol Data Unit Protected Header**

- **Length:** This is a length indicator field that contains the length of the protected header in octets.

The remaining fields are the same as in clear header. These fields fall under an integrity check computation and then optionally under encryption.

**Protected Data**

This information is the original IP packet, which serves as the payload for S-QoS PDU. This payload fall under an integrity check computation and then optionally under encryption.
ICV

This field contains an integrity check value. The ICV is computed over protected header and the payload sections of the PDU. The confidentiality service is provided by encrypting the protected header, payload, and ICV sections of the packet.

The S-QoS entity has access to a S-QoS_MIB. Associated with each label in the S-QoS_MIB are a number of QoS attributes describing how the PDU is to be treated. In response to S-QoS UNITDATA Request, the S-QoS layer constructs a protected data unit and then uses appropriate keys from the MIB to encrypt and/or compute integrity check value (ICV) for the protected data. The resulting PDU is then passed down to the data-link layer. At the receiving end, the S-QoS layer identifies the traffic type and the service desired from the label identified in the label field of the PDU. It also identifies the corresponding key from the MIB.

11.4.5 Security Services

If the security option is enabled, the S-QoS layer provides a connection-oriented integrity service and optionally a connection-oriented confidentiality service. The VBRrt type traffic, as mentioned above, is sensitive to delays. Therefore, to reduce overheads, the S-QoS layer provides an option for connectionless integrity by eliminating the *SEQ. NO.* field. S-QoS layer does not explicitly cater to data origin authentication. However, by using appropriate security related keys and by providing protection to IP addresses of source in the payload field a limited form of data origin authentication is provided. The support for access control is provided using the key service external to S-QoS layer. This key service may choose, for administrative reasons, not to make a key available for communication between any two S-QoS enabled entities. In the case of NTDR networks, an authority that is able to enforce such a rule is the clusterhead. This decision by the key service implies an access control service at the upper interface of S-QoS layer, since S-QoS layer will not allow communication if an appropriate key is not present. In addition, S-QoS will make an access control check against the label and SAI field in the data packet.

11.5 Management Information Base (MIB)

The S-QoS_MIB can be used to store quality of service and security related attributes for each association maintained within the network. It is part of the Management Information Base (MIB). This MIB can be implemented as a table of entries, one for each communicating pair of hosts. It allows the network management applications to control the operation of S-QoS layer. The S-QoS layer implements QoS by manipulating the S-QoS Protocol Data Units (PDUs) according to the QoS attributes stored in the MIB. Appropriate specification and selection of attributes in the S-QoS_MIB enables different degrees of services that can be achieved and availed. These manipulations are completely transparent to the upper protocol layers in that neither the format nor the semantics of the packet are modified. Any IP packet passing through the S-QoS layer is protected and passed on to the data link layer. Similarly, the received packets pass through the S-QoS layer before entering the IP layer.
There is a S-QoS_MIB associated with the S-QoS layer for each node (refer to figure 22). There is a network management entity located in the clusterhead with the corresponding S-QoS_MIB. When a node requests for a communication path with a set of QoS attributes, the clusterhead sets the attributes required for the communication and if the path is established, it updates the MIBs at both its as well as at the requesting node’s end. A typical MIB entry will contain the following information amongst others. The objective here is not to provide the complete MIB specifications but only such information that will help in the understanding of the protocol operation of the S-QoS layer.

![Figure 22: Placement of Management Information Base in NTDR Architecture](image)

Please refer to Appendix C for additional information on the MIB specification.

11.6 Scheduling the Traffic Flows

The participating nodes must schedule traffic flows according to the quality of service parameters specified for the flows. As mentioned before, there are two classes of traffic flows: CBR and VBR. Different scheduling schemes are applied for these two traffic classes. Some scheduling schemes for handling VBR and CBR traffic in wireless environment are suggested in [Sevanto+ 98]. These schemes are equally applicable in the ad-hoc environment. We shall now present an overview of how these scheduling schemes work.

A node maintains a list of scheduling slots for CBR traffic. For each traffic flow, the scheduler allocates slots from the scheduling list according to the bandwidth requirements of the flow. QoS requirements of VBR channels may contain both the minimum allocation requirement (bandwidth requirement) as well as the variable part. The slots from scheduling list are allocated for those VBR channels having bandwidth requirements. The schedulers try to distribute the slots of one flow to the scheduling list as evenly as possible in order to minimise delay variance. For control messages (such as routing messages) predefined percentages of the real bandwidth available will be allocated.
Each slot in the scheduling list accommodates maximum packet size. Schedulers scan the scheduling list in round robin (R-R) order. The principle is that if the currently examined slot has been allocated to a data flow and if that flow has outgoing data waiting in the buffers, the scheduler sends one packet of data for that channel. On the other hand, if the slot has not been allocated to any flow, the scheduler uses the VBR scheduling scheme to choose the next flow. This guarantees maximum link utilization without disturbing the VBR traffic.

In the VBR scheduling scheme, the scheduler uses the prioritised lists, in addition to the slotted scheduling list used for CBR scheme. One scheduling list exists for each priority class. Whenever a new VBR channel is allocated, the scheduler adds the flow to the proper list corresponding to the priority of the flow. New flows are always added to the end of the priority list. When the scheduler reaches an empty slot in the CBR scheduling list, it always checks first in the VBR list with the highest priority. Within each priority class, the channels are scheduled in the round-robin fashion.

11.7 Summary

Providing QoS related features in an ad-hoc environment is a challenging task. Factors such as mobility of hosts, bandwidth and power limitation and unreliable wireless links impose constraints make QoS guarantees difficult to achieve. In this chapter we have made an attempt to identify issues in the provision of QoS in an ad-hoc network environment. We have then attempted to define a dedicated layer for the provision of QoS and security. In particular, we discuss the functionality of this layer, the options available for placement of this layer in the protocol stack and its operational requirements.
Mobile IP protocols are a significant step towards allowing nomadic Internet use. Mobile ad-hoc networks go a step further by providing convenient infrastructure-free communication over the shared wireless channel. However, the presence of mobility in both infrastructure as well as in infrastructure-free environment introduces higher security risks than in a traditional wired network environment. The goal of this thesis was to make a meaningful contribution towards the design of a security framework in mobile and ad-hoc networking systems. Apart from the security related issues, it also addressed the problem of location management in mobile networks and the provision of QoS in ad-hoc networks. In particular, this thesis has contributed in the following areas:

**Location Management in Mobile IP Based Networks**: This thesis contributes by proposing a new framework for supporting mobility in networks. Unlike other major contributions made in this area, our scheme offers some unique features by means of which optimum routing can be established between two communicating mobile hosts. For instance, in our scheme, even though we may have a dedicated home network associated with a roaming host, this network does not play a critical role in formulating routing decisions for the mobile host. The location information pertaining to a moving host is distributed and hence there is no single point of failure.

**Security Extensions to Location Management**: We proposed a security framework for use in distributed location management model (DLM). In the design of this framework we addressed issues relating to key management and public key infrastructure requirements necessary to support the end-to-end-secure communication protocols between two mobile hosts. These protocols are responsible for authentication as well as for the establishment of shared session key between two mobile hosts.

**Provision of secure multicast service in Mobile IP based Networks**: Mobility raises several issues for IP multicast. These issues when combined with the low bandwidth and higher bit error rate of the wireless link make efficient IP multicast a challenging task in a mobile environment. Also, the introduction of mobility further aggravates security problems in multicast based communication. This area of mobile IP multicasting is relatively new and there is not a widely accepted method for multicasting in such environments. This thesis contribution includes a secure multicast framework for IETF mobile IP and Columbia Mobile IP schemes. We have also considered secure group key generation and distribution and anonymity related issues in this context.

**Securing Mobile Ad-hoc Networks**: Security for mobile ad-hoc networks is a new and challenging area that has not been adequately addressed. While there are solutions in place that address different threats, there is no single framework that provides a comprehensive security solution. This thesis contributes by providing a complete security infrastructure for ad-hoc networks that includes the provision of a wide range of security services along with key management and anonymity features. In particular, this thesis has suggested approaches to secure cluster based ad-hoc networks as well.
as an on demand routing protocol AODV. To achieve this objective, a security layer
was designed. Issues relating to the placement of the layer as well as the layer’s
functions and operations are specified.

Provision of Quality of Service in Mobile Ad-hoc Networks: Provision of Quality
of Service features in ad-hoc networks is a relatively new and challenging area that is
currently receiving a lot of attention. QoS based solutions for wired networks do not
scale well for wireless ad-hoc networks due to their peculiar nature of having no fixed
wireless infrastructure and dynamic topology. This thesis contributes by defining a
specification for Quality of Service in the form of a dedicated layer that caters to both
generic QoS features as well as security. In particular this thesis discusses the
functionality of this layer, the options available for placement of this layer in the
protocol stack and its operational requirements.

Future Work: Our investigation did not specifically address denial of service attacks
in ad-hoc networks. Given that ad-hoc networks must often be deployed in hostile
environments and given that any vulnerabilities in them could potentially be exploited
to launch attacks on the global Internet, there exists a compelling need to develop
techniques for protecting these networks against denial of service attacks.

In order to provide a complete range of security services as well as to put in a strong
resistance to denial of service attacks, provision of firewall functionality in ad-hoc
networks must be considered. In a traditional wired network, a firewall is placed in
the ingress/egress point of the network to filter out traffic originating from outside the
boundary of the protected network. However, in an ad-hoc network, the nodes could
be potentially mobile and therefore the topology itself is dynamic. Hence in such
situations there no notion of a well defined ingress/egress point for the network.
Furthermore a traditional firewall is not designed to protect against spoofed packet
flooding attacks wherein the attacker sends legitimate packets that penetrate the
access control rules set by the firewall. Therefore it is worth investigating the design
of a fully distributed, dynamically configurable firewall mechanism is necessary to
prevent the disruption attacks from occurring in an ad-hoc environment.

It is expected that in future, the use of ad-hoc networks in commercial environment
will increase rapidly. For the widespread deployment of such networks it is necessary
to consider the provision of QoS related services in such networks. We have proposed
a QoS based layer designed specifically for mobile ad-hoc networks. However, we
have not defined explicit signalling messages such as those necessary to manage the
label space. The design of a label distribution protocol to support our QoS framework
is a worthwhile task. Finally, provision of security services for non-cluster based ad-
hoc networks and table-driven routing protocols need to be investigated.
Appendix A

TCP/IP and IP Addressing

1. TCP/IP: An Introduction

The TCP/IP protocol suite, the cornerstone of Internet networking, is a four-layer protocol system. Each layer is responsible for a specific task. It is a standard, routable enterprise networking protocol that is the most complete and accepted protocol available. All modern operating systems offer TCP/IP support, and most large networks such as Internet rely on TCP/IP for much of their network traffic.

The four layers, from top to bottom, are the application layer, transport layer, network layer, and link layer. The application layer handles the details of the particular application. This is where applications gain access to the network. There are many standard TCP/IP utilities and services at the application layer such as FTP [RFC 959], Telnet [RFC 854], SNMP versions one, two, and three [RFC 1157, RFC 1902, RFC 2262], and DNS [RFC 1034]. The transport layer protocols provide a communication session between computers. The desired method of data delivery determines the transport protocol. The two transport protocols are Transmission Control Protocol (TCP) [RFC 793] and User Datagram Protocol (UDP) [RFC 768]. TCP provides connection-oriented, reliable communications for applications that typically transfer large amounts of data at one time or that require an acknowledgement for data received. On the other hand, UDP provides connectionless communications and does not guarantee that packets will be delivered. Applications that use UDP typically transfer small amounts of data at one time. Reliable delivery is the responsibility of the application. The motivation behind UDP is that if the amount of data being delivered is small, the overhead of creating connections and ensuring reliable delivery (as in TCP) may be greater that the work transmitting the entire data set.

The network layer protocols encapsulate packets into Internet datagrams and run all the necessary routing protocols. The four network layer protocols are: Internet Protocol (IP) [RFC 791], Address Resolution Protocol (ARP) [RFC 826], Internet Control Message Protocol (ICMP) [RFC 792], and Internet Group Management Protocol (IGMP) [RFC 1112].

- Internet Protocol (IP): A connectionless protocol primarily responsible for addressing and routing packets between hosts and networks. Connectionless means that a session is not established before exchanging data. IP is unreliable in that delivery is not guaranteed. It will always make a “best effort” attempt to deliver a packet. Along the way, a packet might be lost, delivered out of sequence, duplicated, or delayed.
- Address Resolution Protocol (ARP): A protocol used to obtain hardware addresses of hosts located on the same physical network. ARP uses local broadcast of the destination IP address to acquire the hardware address of the destination host or gateway. Once the hardware address is obtained, both the IP address and hardware address are stored as one entry in the ARP cache. The ARP
cache is always checked for an IP address/hardware/address mapping before initiating an ARP request.

- **Internet Control Management Protocol (ICMP):** A protocol responsible for sending messages and reporting errors regarding the delivery of a packet. It should be noted that ICMP does not attempt to make IP a reliable protocol. It merely attempts to report errors and provide feedback on specific conditions. ICMP messages are carried as IP datagrams and are therefore unreliable.

- **Internet Group Management Protocol (IGMP):** A protocol used by IP hosts to report host group memberships to local multicast routers. Specifically, it informs routers that hosts of a specific multicast group are available in a given network. This information is also passed to other routers so that each router that supports multicasting is aware of which host groups are on which network. IGMP packets are carried by IP datagrams and are therefore unreliable.

The overall protocol stack is a tightly coupled system [Perkins 98]. Each layer provides some services that the upper layers use. Thus, support for mobility is likely to affect all the layers. For example, the link layer needs to make provisions to accommodate the distinguishing characteristics of wireless media like low bandwidth and difference in power levels of end-to-end nodes. The network layer that routes data to a destination host based on its location, needs to be modified so that it can handle routing when the physical location of the host changes. Similarly, at the transport layer, it is necessary to provide a better end-to-end delivery service, especially in the case of dropped packets; packets may be lost during mobility and need to be delivered immediately to the new location. Finally, the application layer requires additional support in terms of automatic configuration, service discovery, and link awareness. As an example of an application layer change, if an FTP session is in progress during mobility, the FTP application needs to configure itself being aware of the location changes.

2. Introduction to IP Version 4 Addressing and Routing

In this section we shall discuss two vital features of the IP protocol, namely addressing and routing. An overview of these two aspects of IP protocol is provided, as it is critical for an understanding of Mobile IP based protocols discussed in this thesis.

2.1 IP Addressing

Each TCP/IP host is identified by a logical IP address. A unique IP address is required for each host and network component that communicates using TCP/IP. The IP address identifies a system’s location on the network in the same way a street address identifies a house on a city block. Just as a street address must identify a unique residence, an IP address must be globally unique and have a unique format. Each IP address defines a network identifier (ID) and a host identifier. The network ID identifies the systems that are located on the same physical segment. All systems on the same physical segment must have the same network ID. The network ID must be unique to the Internetwork. The host ID identifies a workstation, a server, a router, or other TCP/IP host within a segment. The address for each host must be unique to the network ID.
Each IP address is 32 bits long and is composed of four 8-bit fields, called octets. Octets are separated by periods (Figure 1). The octet represents a decimal number in the range 0-255. This format is called dotted decimal notation.

![Figure 1: IP Address Syntax [Simeria 95]](image)

IP addresses have been further classified into different classes to accommodate networks of varying sizes.

### 2.2 IP Routing

Routing is the process of choosing a path over which to send the packet. Routing occurs at a TCP/IP host when it sends IP packets and routing occurs in the IP router. A router is a device that forwards packets from one physical network to another. Routers are commonly referred to as gateways. In both cases, the sending host and router, a decision has to be made as to where the packet is to be forwarded. To make these routing decisions, the IP layer consults a routing table that is stored in memory. A routing table contains entries with the IP addresses of router interfaces to other networks that it can communicate with. By default, a router can send packets only to networks to which it has a configured interface.

In a nutshell, the routing process is as follows:

a. When a host attempts communication with another host, IP first determines whether the destination is local or on a remote network.

b. If the destination is remote, IP then checks the routing table for a route to the remote host or remote network.

c. If no explicit route is found, IP uses its default gateway address to deliver the packet to the router.

d. At the router, the routing table is again consulted for a path to the remote host or network. If a path is not found, the packet is sent to the router’s default gateway address.

As each route is found, the packet is sent to the next “hop” router and finally delivered to the destination host. If a route is not found, an ICMP error message is sent to the source host.

The IP protocol uses the various classes of IP addresses to carry out routing process. But currently the Internet has almost exhausted its current set of IP addresses. More recently, to preserve as best as possible the remaining IP address space, routing prefixes have been assigned according to the architecture prescribed in Classless Interdomain Routing (CIDR) [RFC 1518] [RFC 1519]. With CIDR, the network mask is explicitly given separately from the IP address, which allows previous class A and
B networks to be carved up into a much larger number of smaller networks. Furthermore, the smaller networks are usually aggregated so that fewer router advertisements are needed overall at the highest levels of the routing infrastructure.

Routing can be direct or indirect and static or dynamic. One of the problems with the Internet infrastructure is that the number of entries in the routing table has proliferated making table lookups inefficient. There are several techniques available that can make the size of the routing table manageable and handle issues such as security. Please refer to Appendix B for further details on IP routing.

The TCP/IP Protocol Suite uses IP (Internet Protocol) addresses to identify unique locations of network devices in an Internetwork. IP addresses are unique in the sense that each address defines one, and only one, device (host or a router) on the Internet.

3. Structure of IP Addresses

IP addresses are 32 bits long and are written in a "dotted decimal" format (e.g. 192.251.16.72) by separating four 8 bit segments with dots and expressing the value of each segment in decimal. The segments are referred to as "octets" to make clear that they each represent 8 bits.

IP addresses have two parts - a network part and a host part. All addresses on a given network have the same network part but unique host parts, similar to the way houses on a street have unique house numbers but have the same street name in their addresses.

The two part structure of IP addresses, network and host, allows a system of routers to deliver packets to their respective destinations by referring to just the network portion of a packet's destination address. IP Addressing works by translating simple names (such as WWW.IBM.COM) to not-so-simple addresses (like 216.92.81.192). Then the network uses the Classes structure of the IP Address format, comparing it to the network topology (system of routers, hubs and switches), to find a route to relay the data and commands. Each router forwards a packet a step closer to its destination network. Not until a packet reaches its destination network does a router use the host part of the destination address to make final delivery of the packet to its destination device. As an analogy, consider navigating through a city to find a theatre using a street map. After you find the theatres street, one must chart how to get to that street from the current location one intersection at time. When you get to the right street, then it's a matter of finding the specific address of the restaurant on the street.

IP Address Classes

There are five different IP address classes: A, B, C, D, and E. These are designed to cover the needs of different types of organisations. Also, identifying the network portion of an IP address is something the routers must do when forwarding a packet towards its destination. The dots separating octets in the IP address provide some guidance. In class A addresses, the first dot divides the network part from the host part. In class B addresses, the second dot. In class C addresses the third dot. An address is class A if the value of the top octet is between 0 and 127. An address is class B if the value of the top octet is between 128 and 191. An address is class C if the value of the top octet is between 192 and 223. Class D is defined for multicasting. In this class, there is no network id or host id. The whole address represents a group
address. The first four bits define the class (1110). The remaining 28 bits define the multicast group address. Class E is reserved by the Internet for special use. There is no network id or host id in this class. The first four bits define the class (1111).

![Figure 2: IP Addresses: Classes A, B, and C [Simeria 95]](image)

**Subnet Mask**

Most organizations have many more networks than registered network numbers. In fact, many organizations have exactly one registered network number. They divide the address space they are assigned into pieces called "subnetworks" or subnets by assigning special meaning to some of the host bits they have discretion over. If, for example, the registered network number is a Class B (with the last two octets are available for host addresses), the organization will often use the third octet to identify subnetworks and the last octet to identify hosts on a subnetwork. Outside the organization, this subnet structure is not visible or important.

**Classless Inter Domain Routing (CIDR)**

CIDR is a new addressing scheme for the Internet, which allows for more efficient allocation of IP addresses than the old Class A, B, and C address scheme. With a new network being connected to the Internet every 30 minutes the Internet was faced with two critical problems:

- Running out of IP addresses
- Running out of capacity in the global routing tables

Classless Inter-Domain Routing (CIDR) is a replacement for the old process of assigning Class A, B and C addresses with a generalized network "prefix". Instead of being limited to network identifiers (or "prefixes") of 8, 16 or 24 bits, CIDR currently uses prefixes anywhere from 13 to 27 bits. Thus, blocks of addresses can be assigned to networks as small as 32 hosts or to those with over 500,000 hosts. This allows for address assignments that much more closely fit an organization's specific needs.

A CIDR address includes the standard 32-bit IP address and also information on how many bits are used for the network prefix. For example, in the CIDR
address 206.13.01.48/25, the "/25" indicates the first 25 bits are used to identify the unique network leaving the remaining bits to identify the specific host.
Appendix B

IP Routing

1. Introduction

Routing is the process of choosing a path over which to send the data packets. Routing occurs at a TCP/IP host when the host sends IP packets. A router is a device that forwards the packets from one physical network to another. In both cases, sending host and router, a decision has to be made as to where the packet is to be forwarded. To make these routing decisions, IP layer consults a routing table that is stored in memory. A routing table contains entries with the IP addresses of router interfaces to other networks that it can communicate with. By default, a router can send packets only to networks to which it has configured interface. In general, routing process involves the following steps:

1. When a host attempts communication with another host, IP first determines whether the destination host is local or on a remote network.
2. If the destination host is remote, IP then checks the routing table for a route to remote host or remote network.
3. If no explicit route is found, IP uses its default gateway address to deliver the packet to a router.
4. At the router, the routing table is again consulted for a path to the remote host or network. If a path is not found, the packet is sent to the router’s default gateway address.

As each route is found, the packet is sent to the next hop router and finally delivered to the destination host. If a route is not found, an error message is sent to the source host.

2. Direct versus Indirect Delivery

The delivery of a packet to its final destination is accomplished using two different methods of delivery: direct and indirect.

In a direct delivery, the final destination of the packet is a host connected to the same physical network. Direct delivery occurs when the source and destination of the packet are located on the same physical network or if the delivery is between the last router and the destination host. The sender can easily determine if the delivery is direct by extracting the network address of the destination packet (setting the host id part to all 0s) and compare this address with the addresses of the networks to which it is connected. If a match is found, the delivery is direct.

If the destination host is not on the same network as the deliverer, the packet is delivered indirectly. In an indirect delivery, the packet goes from router to router until it reaches the one connected to the same physical network as its final destination.

Note that a delivery always involves one direct delivery but zero or more indirect deliveries. Note also that the last delivery is always a direct delivery.
3. Static versus Dynamic Routing

Static routing is a function of IP. Static routers require that routing tables are built and updated manually. If a route changes, static routers do not inform each other of the change, nor do static routers exchange routes with dynamic routers.

Dynamic routing is a function of routing protocols, such as the Routing Information Protocol (RIP) and Open Shortest Path First (OSPF). Routing protocols periodically exchange routes to known networks among dynamic routers. If a route changes, other routers are automatically informed of the change.

4. Routing Methods

Routing, as mentioned before, requires a host or a router to have a routing table. When a host has a packet to send or when a router has received a packet to be forwarded, it looks at this table to find the route to the final destination. However, this simple solution is impossible today in an internetwork such as the Internet because the number of entries in the routing table make table lookups inefficient. Several techniques can make the size of the routing table manageable and handle such issues as security. We will briefly discuss these methods here.

4.1 Next Hop Routing

In this technique, the routing table holds only the addresses of the next hop instead of holding information about the complete route. Routing tables are thereby consistent with each other.

4.2 Network Specific Routing

In this technique, instead of having an entry for every host connected to the same physical network, we have only one entry to define the address of the network itself. In other words, this technique treats all hosts connected to the same network as one single entity. This technique not only makes the routing table smaller but also simplifies the search process.

4.3 Host-Specific Routing

In host specific routing, the host address is given in the routing table. The idea of host-specific routing is the inverse of network-specific routing. Here efficiency is sacrificed for other advantages: Although it is not efficient to put the host address in the routing table, there are occasions in which the administrator wants to have more control over routing. For example, if an administrator wants all packets arriving for a host B delivered to router R3 instead of router R1, one single entry in the routing table of host A can explicitly define the route.

4.4 Default Routing

This is another technique to simplify the routing process. For example consider a host A connected to a network with two routers. Router R1 is used to route the packets to hosts connected to network N2. However, for the rest of the Internet, router R2 should
be used. So instead of listing all networks in the entire Internet, host A can just have
one entry called the default (network address 0.0.0.0).
Appendix C

Introduction to IP Multicast

1 Introduction

Multicast IP was developed to facilitate next generation applications that can rely on the efficient delivery of packets to multiple destinations across an internetwork [Maufer 98]. Multicast falls between unicast and broadcast methods of message delivery. The unicast mode involves delivery from one source to one destination. At the other extreme, broadcast mode involves indiscriminate delivery from one source to all destinations. Multicast aims to deliver packets to subnetworks only if they contain end stations that have requested to receive the traffic. Broadcasting has a major disadvantage: its lack of selectivity. If an IP datagram is broadcast to a subnet, every host on the subnet will receive it, and have to process it to determine whether the target protocol is active. If it is not, the IP datagram is discarded. Multicasting avoids this overhead by using groups of IP addresses.

Multicasting techniques are being increasingly used by applications to disseminate data across the network. Some applications that make use of multicasting include videoconferencing, audio conferencing, shared whiteboards and distributed interactive simulations. The main motivation for deploying a multicasting infrastructure is for efficiency reasons. Today there is a critical mass of multicast routing protocols available from major vendor routers that should meet the requirements of most intranets. Also, most modern end-station IP stacks support multicast sending and receiving. Due to incredible market penetration of web browsers, a popular browser plugin that operates over multicast would generate absolutely enormous demand for multicast IP in Intranets all over the work [Maufer 98]

IP multicast [Deering+ 90] provides a popular model of multicast over the Internet. Each multicast group is identified by a group address. Members can join and leave the group using a multicast protocol such as [RFC 1112]. IP packets addressed to a group address are delivered using a multicast tree to all group members. The sender does not need to know the membership of the group. Current implementations of IP multicast in the Internet use either source specific multicast distribution tree or group shared tree. In the source specific multicast tree each sender of a group has its own multicast tree rooted at the sender whereas in the group-shared tree, each member of the group must explicitly join a predefined multicast delivery tree.

2. Multicast Addressing

A 28-bit number represents each group, which is included in a Class D address. Recall that a class D address has the format (Figure 10):
Figure 1: IP Multicast Class D Address Format

So multicast group addresses are IP addresses in the range 224.0.0.0 to 239.255.255.255. For each multicast address there is a set of zero or more hosts, which are listening to it. This set is called the host group. There is no requirement for any host to be a member of a group to send to that group. There are two kinds of host group:

Permanent

The IP address is permanently assigned by the Internet Assigned Numbers Authority (IANA). The membership of a host group is not permanent: a host may leave or join the group at will. The list of IP addresses assigned permanent host groups is included in STD 2 - Assigned Internet Numbers. Important ones are:

- 224.0.0.0: Reserved base address
- 224.0.0.1: All systems on this subnet
- 224.0.0.2: All routers on this subnet

Some other examples used by the OSPF routing protocol (see Open Shortest Path First Protocol (OSPF) Version 2) are:

- 224.0.0.5: All OSPF routers
- 224.0.0.6: OSPF Designated Routers

An application may also retrieve a permanent host group's IP address from the domain name system using the domain mcast.net, or determine the permanent group from an address by using a pointer query in the domain 224.in-addr.arpa. A permanent group exists even if it has no members.

Transient

Any group, which is not permanent is transient and is available for dynamic assignment as needed. Transient groups cease to exist when their membership drops to zero.

Address Resolution

Multicasting on a single physical network, which supports the use of multicasting is simple. To join a group, a process running on a host must somehow inform its network device drivers that it is wishes to be a member of the specified group. The device driver software itself must map the multicast address to a physical multicast address and enable the reception of packets for that address. The device driver must also ensure that the receiving process does not receive any spurious datagrams by checking the destination address in the IP header before passing it to the IP layer.

For example, Ethernet supports multi-casting if the high-order byte of the 48-byte address is X'01' and IANA owns an Ethernet address block, which consists of the addresses between X'00005E000000' and X'00005EFFFFFF'. The lower half of this
range has been assigned by IANA for multicast addresses, so on an Ethernet LAN there is a range of physical addresses between X'01005E000000' and X'01005E7FFFFF' which is used for IP multicasting. This range has 23 usable bits, so the 28-bit multicast addresses are mapped to the Ethernet addresses by considering the low-order 23 bits, that is 32 multicast addresses are mapped to each Ethernet address. Because of this non-unique mapping, filtering by the device driver is required. There are two other reasons why filtering might still be needed:

- Some LAN adapters are limited to a finite number of concurrent multicast addresses and if this is exceeded they must receive all multicasts.
- Other LAN adapters tend to filter according to a hash table value rather than the whole address, which means that there is a chance that two multicast addresses with the same hash value might be in use at the same time and the filter might ``leak''.

Despite this requirement for software filtering of multicast packets, multicasting still causes much less overhead for hosts that are not interested. In particular, those hosts that are not in any host group are not listening to any multicast addresses and all multicast messages are filtered by the network interface hardware.

3. Multicast Protocols

Multicasting is not limited to a single physical network. It is possible for members of the group to be spread across separate physical networks. In this case, special multicast gateways forward multicast datagrams. There are two aspects to multicasting across physical networks:

- Hosts need to join a multicast group.
- There should be an appropriate protocol in routers to forward Multicast datagrams to the recipients.

To help solve the first problem, the Internet Group Management Protocol (IGMP) [RFC 1112] was designed. IGMP has been designed to help a multicast router identify hosts in a LAN that are members of a multicast group. It is a companion to the IP protocol. To help the second problem, various multicast routing protocols such as Distance Vector Multicast Routing Protocol (DVMRP) [RFC 1075], and Protocol Independent Multicast –Sparse Mode [RFC 2362] and dense [Deering+ 97] modes have been designed. We shall now provide an overview of these multicast protocols.

3.1 Internet Group Management Protocol (IGMP)

Before a multicast router can propagate multicast membership information, it must determine that one or more hosts on the local network have decided to join a multicast group. To do so, multicast routers and hosts that implement multicast must use IGMP [RFC 1112] to communicate group membership information. IGMP is a signalling protocol and is only locally significant. Routers do not forward IGMP packets. Besides allowing end stations to join and maintain membership in multicast groups, IGMP allows routers to explicitly query subnetworks for members of any group.

Conceptually, IGMP has two phases:
**Phase One:** When a host joins a new multicast group, it sends an IGMP message to the group’s multicast address declaring its membership. Local multicast routers receive the message, and establish necessary routing by propagating the group membership information to other multicast routers using multicast routing protocols.

**Phase Two:** Because membership is dynamic, local multicast routers periodically poll hosts on the local network to determine whether any hosts still remain members of each group. If any host responds for a given group, the router keeps the group active. If no host reports membership in a group after several polls, the multicast router assumes that none of the hosts on the network remain in the group, and stops advertising group membership to other multicast routers. While IGMP queries are sent to all hosts in a subnet, IGMP reports are only sent to the concerned group members.

IGMP is an integral part of IP and has been designed, deployed, and improved in an iterative cycle over the last decade or so. In a sense, IGMP is the most important protocol relative to IP multicast: without it, nothing else works. There have been two versions of IGMP so far. Usage of IGMP v1 [RFC 1112] has probably all but died out and all modern IP multicast implementations support IGMP v2 [Fenner 97] by default. The features of version 1 are still present in version 2, with some enhancements to improve efficiency.

**3.2 Multicast Routing Paradigms**

**3.2.1 Introduction**

The primary means of classifying multicast routing protocols is by noticing how senders communicate with receivers. The basic job of a multicast routing protocol is to build a distribution tree so that traffic may be efficiently delivered to all the active receivers, which are members of the group, from any source. There are two common types of distribution trees: “broadcast and prune”, and “shared trees”.

Broadcast and Prune oriented protocols build source-based trees. These protocols make a key assumption that the multicast data traffic itself is a convenient way to control the construction of the tree. Hence they are also called “data-driven” protocols. As a result, periodically the data is reliably broadcast from the source to the extremities of the internetwork in search for active group members.

In broadcast and prune protocols, the first packet for a new multicast data flow is broadcast to the edges of the internetwork. At the internetwork’s extremities, the edge routers consult their IGMP derived local group databases. If there are no local members of the group, the router may remove itself from the delivery tree by sending prune messages upstream. As a result, the tree only includes branches that lead to active group members. Such branches are called active branches. The prune lifetime is kept very short. Once the prunes expire, any traffic for multicast flows that have no prune state is rebroadcast to the edges of the internetwork. There is an explicit message by means of which it is possible to join (graft) back to a previously pruned branch of the group’s delivery tree. When an upstream node receives a graft message, it cancels out the previous prune message. Graft messages cascade hop by hop toward
the source until they reach the nearest “live” branch point on the delivery tree. In this way, previously pruned branches are quickly restored to a given delivery tree. Examples of broadcast and prune protocols are Distance Vector Multicast Routing Protocol (DVMRP) [RFC 1085] and Protocol Independent Multicast – Dense Mode (PIM-DM) [Deering+ 97].

Shared Trees on the other hand have a central point to which all receivers attach. The central point distributes traffic sent to pre-registered group members. All traffic destined for the group emanates from this central point. The key difference between the data-driven broadcast and prune protocols and shared tree protocols is that shared trees must be joined prior to the flow of data; otherwise, the central point knows of no downstream receivers, and data stops there. Inside a shared tree routing domain, it is possible that there is only one centre serving all the groups, or they may be multiple centres, each in use by a different set of groups. However, each group receives its traffic over the same delivery tree, regardless of the source. Example of shared tree protocol is Protocol Independent Multicast – Sparse Mode (PIM-SM) [RFC 2362]

Some popular multicast protocol implementations under broadcast and prune shared tree based approaches are discussed below. Besides these protocols, multicast extensions have been defined for Open Shortest Path First (OSPF) routing protocols. OSPF routing protocol is defined in [RFC 2178]. The multicast extensions for OSPF is defined in [RFC 1584]. We provide a brief description of this protocol.

### 3.2.2 Distance Vector Multicast Routing Protocol (DVMRP)

DVMRP uses a technique known as Reverse Path Forwarding. When a router receives a packet, it floods the packet out of all paths except the one that leads back to the packet's source. Doing so allows a data stream to reach all LANs (possibly multiple times). If a router is attached to a set of LANs that do not want to receive a particular multicast group, the router can send a "prune" message back up the distribution tree to stop subsequent packets from travelling where there are no members.

DVMRP will periodically re-flood in order to reach any new hosts that want to receive a particular group. There is a direct relationship between the time it takes for a new receiver to get the data stream and the frequency of flooding. DVMRP implements its own unicast routing protocol in order to determine which interface leads back to the source of the data stream. This unicast routing protocol is very like RIP and is based purely on hop counts. As a result, the path that the multicast traffic follows may not be the same as the path that the unicast traffic follows. The multicast datagram has a Time To Live (TTL) value, which is decremented with each hop to a new network. When the Time to Live field is decremented to zero, the datagram can go no further.

DVMRP has significant scaling problems. Like RIP, DVMRP uses a small value for infinity. More important, the amount of information DVMRP keeps is overwhelming: in addition to entries for each active (group, source), it must also store entries for previously active groups so it knows where to send a graft message when a host joins a group that was pruned. Finally, DVMRP uses a broadcast and prune paradigm that generates traffic on all networks until membership information can be propagated. Ironically, DVMRP also uses a distance vector algorithm to propagate membership
information, which makes propagation slow. To overcome the limitations of DVMRP, the IETF has investigated other multicast protocols. Efforts have resulted in several designs including Protocol Independent Multicast (PIM) – Sparse mode [RFC 2362] and dense mode [Deering+ 97], and Multicast Extensions to OSPF (MOSPF) [RFC 1584]

3.2.3 Protocol Independent Multicast (PIM)

Introduction

The Inter-Domain Multicast Routing (IDMR) working group of the IETF developed the Protocol-Independent Multicast routing protocols. The objective of the IDMR working group is to develop one or possibly more than one standards-track multicast routing protocols that can provide scalable multicast routing across the Internet.

PIM receives its name because it is not dependent on the mechanisms provided by any particular unicast routing protocol. As mentioned before, DVMRP has a built-in unicast routing protocol to facilitate locating source subnetworks relative to any router, and to help determine which interfaces were children with respect to any source.

PIM consists of two independent protocols that share little beyond the name and basic message header formats: PIM – Dense Mode (PIM-DM) and PIM-Sparse Mode (PIM-SM). The distinction arises because no single protocol works well in all possible situations. We shall briefly examine these protocols.

PIM-Dense Mode (PIM-DM)

PIM-DM is designed for a LAN environment in which all, or nearly all, networks have hosts listening to each multicast group. Hence, PIM-DM can be deployed in resource-rich environments such as campus LANs, where group members densely occupy a large percentage of the subnetworks, and bandwidth is likely to be readily available.

Dense-Mode PIM builds source-based trees based on the reverse path multicasting (RPM) algorithm as in DVMRP. A minor difference from DVMRP is that PIM-DM simply forwards multicast traffic on all non-incoming interfaces until explicit prune messages are received, unlike the DVMRP, which calculates a set of child interfaces for each source pair. PIM-DM trades off a bit of extra flooding for a simpler protocol design and to avoid the overhead inherent in determining parent-child relationships.

Pruning PIM-DM only happens via explicit prune messages, which are multicast on broadcast links. Other routers present may hear this prune message and still wish to receive traffic for this group to support their own active downstream receivers; these other routers must multicast PIM-Join packets to ensure they remain attached to the distribution tree. PIM-DM uses a reliable graft mechanism to enable previously sent prunes to be removed when new downstream group members appear after a prune is sent.
Since PIM-DM uses RPM, it implements a reverse path check on all packets it receives. Again, this check verifies that received packets arrive on the interface, which the router would use if it needed to send a packet toward the source’s subnetworks. Since PIM-DM does not have its own routing protocol (as opposed to DVMRP), it uses existing unicast routing table to orient itself with respect to the source(s) of multicast packets it has seen.

In a nutshell, PIM Dense mode is most useful when:

- Senders and receivers are in close proximity to one another.
- There are few senders and many receivers.
- The volume of multicast traffic is high.
- The stream of multicast traffic is constant.

Using PIM, some duplicate packets may exist in the network. But the cost incurred is the same as DVMRP.

**PIM-Sparse Mode**

PIM-SM is based on the shared tree approach and is described in [RFC 2362]. PIM-SM is a demand driven protocol that needs a point to which Join messages can be sent. Therefore, sparse mode designates a router called a *Rendezvous Point* (RP) that serves as the multicast core router. When a host joins a multicast group, the local router unicasts a join request to the RP, routers along the path examine the message, and if any router is already a part of the tree, the router intercepts the message and replies. Thus, PIM-SM builds a shared forwarding tree for each group, and the trees are rooted at the rendezvous point.

PIM-SM has the ability to optimise connectivity through reconfiguration. For example, instead of a single RP, each sparse mode router maintains a set of potential RP routers, with one selected at any time. If the current RP becomes unreachable due to a network failure causing disconnection, then PIM-SM selects another RP from the set and starts rebuilding the forwarding tree for each multicast group.

In addition to selecting an alternative RP, PIM-SM can switch from the shared tree to a Shortest Path Tree (SPT) mode. The motivation is due to the fact that although the shared tree approach forms shortest paths from each host to the RP, it may not optimise routing. In particular, if group members are not close to the RP, the inefficiency can be significant. Therefore, PIM-SM includes a facility to allow a router to choose between the shared tree or a shortest path tree to the source sometimes called the source tree. Although switching trees is conceptually straightforward, many details complicate the protocol. For example, most implementations use the receipt of traffic to trigger a change. If the traffic from a particular source exceeds a preset threshold, the router begins to establish a shortest path. Unfortunately, traffic can change rapidly, so routers must apply hysteresis to prevent oscillations. Furthermore, the change requires routers along the shortest path to cooperate; all routers must agree to forward the datagrams for the group. Also, because the change affects only a single source, a router must continue its connection to the shared tree so it can continue to receive from other sources. More important it must keep sufficient routing information to avoid forwarding multiple copies of each
datagram from a (source, group) pair for which a shortest path tree is established. In a nutshell, sparse multicast is most useful when:

- There are few receivers in a group.
- Senders and receivers are separated by WAN links.
- The type of traffic is intermittent.

However, some issues concerning PIM-SM are still not resolved. It is a router-to-router protocol, and therefore all routers in the network must be upgraded to support it. Another problem is the location of the RP. There are many ISPs in the world, and none of them want to depend on an RP in the domain of another ISP for multicast service between its own customers.

### 3.2.4 Multicast Extensions to OSPF (MOSPF)

Multicast OSPF (MOSPF) was defined as an extension to the OSPF unicast routing protocol in [RFC 1584]. OSPF works by having each router in a network understand all of the available links in the network. Each OSPF router calculates routes from itself to all possible destinations.

MOSPF works by including multicast information in OSPF link state advertisements. An MOSPF router learns which multicast groups are active on which LANs. MOSPF builds a distribution tree for each source/group pair and computes a tree for active sources sending to the group. The tree state is cached, and trees must be recomputed when a link state change occurs or when the cache times out. MOSPF works only in internetworks that are using OSPF.

MOSPF is best suited for environments that have relatively few source/group pairs active at any given time. It will work less well in environments that have many active sources or environments that have unstable links.
Appendix D

S-QoS MIB

Label
Source Address
Destination Address
Source Port Number
Destination Port Number
SE
SAI

Traffic_Desc_Para

(VBR). CBR
VBRrt
VBRnrt
PIR
Loss_Ratio
ABR
Sec-Options

CO-C (Connection-oriented Confidentiality) and CO-I (Connection-oriented Integrity)
CO-I
CLS-C (Connectionless Confidentiality) and CLS-I (Connectionless Integrity)
CLS-I

Sec_Para

Confidentiality

• ConfidentialityAlgorithmBlockSize Integer32,
• ConfidentialityAlgorithmDecryptKeyLength Integer32,
• ConfidentialityAlgorithmDecryptKey OCTET STRING,
• ConfidentialityAlgorithmEncryptKeyLength Integer32,
• ConfidentialityAlgorithmEncryptKey OCTET STRING,
• ConfidentialityAlgorithmInitVectorIndicate TruthValue,
• ConfidentialityAlgorithmInitVectorLength Integer32,
• ConfidentialityAlgorithmInitVector OCTET STRING,
• ConfidentialityAlgorithmOperateMode DisplayString,
• ConfidentialityAlgorithmSymmetricIndicate TruthValue,

Integrity

• DigestAlgorithmInitVectorIndicate TruthValue
• DigestAlgorithmInitVectorLength Integer32,
• DigestAlgorithmInitVector OCTET STRING
• IntegrityAlgorithmInputSize
• IntegrityAlgorithmOutputSize
• SignatureAlgorithmIdentifier
• SignatureAlgorithmInitVectorIndicate TruthValue
• SignatureAlgorithmInitVectorLength Integer32
• SignatureAlgorithmInitVector
• SignatureAlgorithmInputSize
• SignatureAlgorithmOutputSize
• SignatureAlgorithmSymmetricIndicate

Authentication

• Certificate of Source
• Certificate of Destination
• Nonces
• Timestamps
• Session Key lifetime.

Audit

• PDUs discarded due to failed integrity check.
• PDUs discarded due to missing access control label.
• PDUs discarded due to mismatch in label.
List of Publications


R. Shankaran, V. Varadharajan and M. Hitchens “Secure Multicast Support for Mobile IP with Hierarchical Registration Approach” Proceedings of the 14th IASTED International Conference Parallel and Distributed Computing and Systems (PDCS), November 4-6, 2002, Cambridge, USA.


References


between Systems - Local and Metropolitan Area Network - Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications 1999.


Security Issues in Mobile IP and Mobile Ad Hoc Networks

A thesis submitted in fulfilment of the requirements for the award of
the degree

Doctor of Philosophy

From

University of Western Sydney

By

Rajan Shankaran

School of Computing and Information Technology (CIT)

NOVEMBER 2004
Dedicated to the Lotus Feet of H.H. Shree Mataji Nirmala Devi
## CONTENTS

Declaration  
Acknowledgments  
List Of Figures  

1. INTRODUCTION  
   1.1 Introduction  
   1.2 Thesis Structure  
      1.2.1 Scope  
      1.2.2 Organization  

PART A: MOBILE COMPUTING  
2. INTRODUCTION TO MOBILE COMPUTING  
   2.1 Introduction  
   2.2 Mobile Computing: Definition and Uses  
   2.3 Classifying Mobile Networks  
   2.4 Design Issues in Mobile Computing  
   2.5 Summary  

PART B: LOCATION MANAGEMENT/SECURITY IN MOBILE IP  
3. INTRODUCTION TO MOBILE IP  
   3.1 Introduction  
   3.2 Next Generation: Internet Protocol Version 6 (IPv6)  
   3.3 Routing in Mobile Environment  
      3.3.1 IETF Mobile IP Architecture  
      3.3.2 Other Three Prominent Mobile IP Schemes  
   3.4 Other Related Work in Location Management  
   3.5 Secure Location Management in Mobile Networks  
      3.5.1 IP Security  
      3.5.2 Standard Mobile IP Authentication  
      3.5.3 Mobile IP Authentication with AAA Infrastructure  
      3.5.4 Public Key Based Secure Mobile IP  
      3.5.5 Secure Mobile IP using Firewall Support  
      3.5.6 Use of IPSec in Mobile IP  
      3.5.7 Location Privacy in Mobile IP  
   3.6 Summary  

4. DISTRIBUTED LOCATION MANAGEMENT SCHEME FOR MOBILE HOSTS  
   4.1 Introduction  
   4.2 Our Architectural Approach  
      4.2.1 Definitions and Assumptions  
      4.2.2 Data Structures maintained by DLM Entities  
      4.2.3 Distributed Location Management Scheme (DLM)  
   4.3 Tunneling Procedures  
      4.3.1 IP-In-IP Encapsulation  


Declaration

This is to certify that the work reported in this thesis was done by the author, unless specified otherwise, and that no part of it was submitted in a thesis to any other University or similar institution.

Date Rajan Shankaran
ACKNOWLEDGMENTS

There are a number of individuals who made this thesis possible, specifically my supervisors Professor Vijay Varadharajan and Dr. Michael Hitchens, who demonstrated an abundance of encouragement, patience and understanding and had to sacrifice a part of them to make this endeavour real.

Professor Vijay Varadharajan has been an essential part of my career. He has taught me much about developing secure systems and protocols and also formed many of my early ideas about research. He has been a constant source of inspiration and helpful advice. He was also instrumental in arranging the much needed financial assistance necessary to support this work. His unrelenting technical feedback has led to the timely completion of this work. I can safely say that without his support this thesis would not have happened. I am grateful to Dr. Michael Hitchens for his critical, comprehensive and constructive feedback, which was truly invaluable. I have benefited a lot from many stimulating discussions with him on the subject of network security.

I would also like to acknowledge the financial support that I have received from Professor Varadharajan’s grant at the University of Western Sydney for the past four and a half years. In addition, I am indebted to Gar Jones for bailing me out of a financially difficult situation during the course of this work. Thanks to Dr. Yan Zhang for reviewing the draft of this thesis and for his helpful suggestions. Barbara Pinning and in general, the CIT school at University of Western Sydney provided me additional support for which I am extremely grateful. This also brings to an end my long association with University of Western Sydney where I have spent the last eight years of my student life. It was an enjoyable experience.

I would also like to thank William Zhao for his encouragement and friendship over the years. I am grateful to Andrina Brennan for her support and assistance. This thesis is far better for her having put in so much effort in proof reading it. She was also helpful in numerous other ways that cannot be expressed in words. My special thanks to Christina Harvey for her understanding and helpfulness during the final stages of submission.

Last but not the least, I gratefully acknowledge the support, encouragement and patience of my family through yet another degree. My brother Anand provided a great source of motivation when it came to facing difficult times in life. His monumental sacrifices for the family have helped me achieve my personal goals. My parents, N. L. Shankaran and Dr. Rajeswari Shankaran, who endured it with infinite patience and good grace. Finally, my wife Surekha who provided a great source of motivation when it came to facing the difficult times in life. Her strong faith in God also helped me to overcome my many difficulties and doubts.

MAY GOD ALMIGHTY BLESS THEM ALL!

Date Rajan Shankaran
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>IPv6 Header Format</td>
<td>26</td>
</tr>
<tr>
<td>Figure 2</td>
<td>IETF Mobile IP Architecture</td>
<td>27</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Tunnelling Procedure in IETF Mobile IP</td>
<td>30</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Reverse Tunnelling in IETF Mobile IP</td>
<td>31</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Columbia Mobile IP Architecture</td>
<td>33</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Sony VIP Option Format</td>
<td>34</td>
</tr>
<tr>
<td>Figure 7</td>
<td>IBM Mobile IP Scheme</td>
<td>36</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Mobile IP AAA Trust Model</td>
<td>47</td>
</tr>
<tr>
<td>Figure 9</td>
<td>A Distributed Location Management (DLM) Architecture</td>
<td>66</td>
</tr>
<tr>
<td>Figure 10</td>
<td>The Autonomous System Implementation of DLM Architecture</td>
<td>74</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Local Registration M-IP System</td>
<td>117</td>
</tr>
<tr>
<td>Figure 12</td>
<td>Columbia Mobile Internetworking</td>
<td>136</td>
</tr>
<tr>
<td>Figure 13</td>
<td>NTDR Network Architecture</td>
<td>184</td>
</tr>
<tr>
<td>Figure 14</td>
<td>NTDR MAC Protocol</td>
<td>186</td>
</tr>
<tr>
<td>Figure 15</td>
<td>Propagation of RREQ and RREP Messages</td>
<td>209</td>
</tr>
<tr>
<td>Figure 16</td>
<td>Extensions to RREQ Message</td>
<td>219</td>
</tr>
<tr>
<td>Figure 17</td>
<td>Extensions to RREP Message</td>
<td>220</td>
</tr>
<tr>
<td>Figure 18</td>
<td>Extensions to RRERR Message</td>
<td>220</td>
</tr>
<tr>
<td>Figure 19</td>
<td>S-QoS Protocol Data Unit</td>
<td>234</td>
</tr>
<tr>
<td>Figure 20</td>
<td>S-QoS Protocol Data Unit Clear Header</td>
<td>234</td>
</tr>
<tr>
<td>Figure 21</td>
<td>S-QoS Protocol Data Unit Protected Header</td>
<td>235</td>
</tr>
<tr>
<td>Figure 22</td>
<td>Placement of Management Information Base in NTDR Architecture</td>
<td>237</td>
</tr>
</tbody>
</table>