Signaling for Secure and Efficient QoS-aware Mobility Support in IP-based Cellular Networks

vorgelegt von
Master of Science
Tianwei Chen
aus Berlin

von der Fakultät IV – Elektrotechnik und Informatik –
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
– Dr.-Ing. –
genehmigte Dissertation

Promotionsausschuß:

Vorsitzender:	Prof. Dr. rer. nat. U. Heiß
Gutachter:	Prof. Dr.-Ing. A. Wolisz
Gutachter:	Prof. Dr. rer. nat. P. Müller

Tag der wissenschaftlichen Aussprache: 20. Dezember 2004

Berlin 2005
D 83

Contents

List of Figures

List of Tables

Zusammenfassung

Im Laufe der letzten Jahre haben sich die Dienstleistungen der Mobilitätskommunikation von einem eingeschränkten Angebot mit spärlicher Abdeckung und begrenzter Mobilität hin zu einer ubiquitär verfügbaren Kommunikationsinfrastruktur entwickelt, die es erlaubt, mit handlichen Multifunktionsgeräten sowohl klassische Telefoniedienste als auch Daten-orientierte Anwendungen in Anspruch zu nehmen. Weiterhin zeichnet sich eine steigende Tendenz zur Netzkonvergenz ab, so dass zukünftig mit einem mobilen Internetzugang auf der Grundlage einer All-IP-basierten Infrastruktur zu rechnen ist, der verschiedene drahtlose Technologien unterstützt und eine nahtlose Interoperabilität zwischen diesen Technologien ermöglicht.

Diese Dissertation adressiert eine zentrale Herausforderung dieser Entwicklung: die Frage wie in einer solchen Netzarchitektur Mobilität, Sicherheit und Dienstgüte (Quality of Service, QoS) bei der Bereitstellung von Diensten gemeinsam gewährleistet werden können. Während eines so genannten Handover (Wechsel der Zugangsstation, in der Regel ausgelöst durch Benutzermobilität) müssen neben der grundlegenden Mobilitätssignalisierung auch Sicherheitsüberprüfungen (d.h. Authentisierung und Autorisierungsprüfung) und Maßnahmen zur Gewährleistung der vereinbarten Dienstqualität durchgeführt werden.

Hierbei kann trotz der Komplexität der zu lösenden Aufgaben lediglich eine minimale Wartezeit in Kauf genommen werden, um Unterbrechungen verzögerungssensitiver Anwendungen zu vermeiden bzw. minimieren. Es ist jedoch dafür Sorge zu tragen, dass eine technische Lösung dieser Fragestellung zentrale Sicherheitsrisiken, wie zum Beispiel Sabotageangriffe (Denial of Service, DoS) auf Zeichengabeinstanzen im Zugangsnetz und Bekanntwerden vertraulicher Benutzerinformationen, angemessen berücksichtigt.

Um diesen Anforderungen gerecht zu werden, wurde in der vorliegenden Dissertation ein Entwurf für eine sichere und leistungsfähige QoS-gerechte Mobilitätsunterstützung erarbeitet, der insbesondere die Mobilität innerhalb von Zugangsnetzen optimiert unterstützt, die innerhalb einer administrativen Domäne liegen. Die Grundidee des Ansatzes ist, Maßnahmen nicht nur in jedem einzelnen Schritt, aber auch in der Optimierung des Zusammenwirkens einzelner Teilabläufe der Prozedur zu ergreifen. Kernpunkte des Ansatzes sind 1.) die Anreicherung der vom Zugangsnetz ausgesendeten so genannten Advertisement-Nachrichten mit Dienstgüteinformationen, um einem Mobilitätsbenutzer zu ermöglichen, den besten Zugangs-Router als das Übergabeziel auszuwählen; 2.) die Trennung der Authentisierung in zwei Teilschritte, eine vorläufige Glaubwürdigkeitsprüfung zur Reduzierung des DoS-Risikos und die eigentliche Überprüfung der Nutzeridentität (bzw. der Identität des verwendeten Geräts); 3.) die Integration der Binding-Update-Signalisierung und der Dienstgütesignalisierung (d.h.

QoS+BU); 4.) die Parallelisierung der QoS+BU-Prozedur und der Authentisierungs- und Autorisierungsprüfung, und 5.) die unmittelbare Fortsetzung der Kommunikationsverbindung nach erfolgter QoS+BU-Signalisierung, wobei zur temporären Absicherung der ausgetauschten Daten bis zum Abschluss der Authentisierungs- und Autorisierungsprüfung eine vorläufige IPSec-Sicherheitsassoziation eingesetzt wird, die nach Abschluss dieser Prüfungen durch eine neue Assoziation mit frisch ausgehandelten Sitzungsschlüsseln ersetzt wird.

In der Arbeit wird die Reduktion des DoS-Risikos durch diesen Ansatz auf der Grundlage eines Kosten-basierten Ansatzes formal analysiert und nachgewiesen. Weiterhin wird die Leistungsfähigkeit des spezifizierten integrierten Zeichengabeprotokolls mittels Warteschlangen-theoretischer Berechnungen untersucht und mit der existierender Ansätze verglichen. Die für diese Analyse benötigten Eingangswerte wurden durch Messungen an einem ebenfalls im Rahmen der Arbeit implementierten Prototyp gewonnen, mit dem gleichzeitig auch die praktische Umsetzbarkeit des Entwurfs erprobt und nachgewiesen wurde.

Die Ergebnisse der Arbeit zeigen, dass der entwickelte Ansatz die Anforderungen an eine sichere und QoS-gerechte Handover-Zeichengabeprozedur erfüllt. Insbesondere zeigen die Ergebnisse die erhöhte Robustheit der Prozedur gegenüber Sabotageangriffen bei gleichzeitiger Einhaltung der einzuhaltenden Leistungsanforderungen. Mit der prototypischen Implementierung und experimentellen Überprüfung wurde darüber hinaus die Umsetzbarkeit des Ansatzes in die Praxis nachgewiesen.

Abstract

In the last several years mobile communication services have evolved from constrained mobile devices, sparse coverage and limited mobility to an almost ubiquitous communication infrastructure. The evolvement allows small-size and multi-functional devices to run both classical telephony and data oriented applications. Furthermore, the growing trend towards network convergence predicts that upcoming ubiquitous mobile Internet access will be realized by an All-IP-based infrastructure that supports a diverse set of wireless technologies and realizes seamless interoperability based on protocols of the IP protocol suite.

The thesis addresses the main challenge of the development: in which network architecture IP mobility featuring security and Quality of Service (QoS) can be conjointly provisioned? During a namely handover procedure (changing of access station being triggered by user's mobility), the operations of security checks (i.e. authentication and authorization), proper QoS guarantee should also be performed in addition to basic mobility management signaling.

Despite of the complexity, it is crucial that performing the operations should introduce minimal latency in order to minimize interruption to the delay sensitive applications. Meanwhile, some security issues such as Denial of Service (DoS) attacks and exposure of a mobile user's confidential information in the visited foreign network should be taken into account.

To meet the requirements, a scheme for a secure and efficient QoS-aware mobility support is proposed in the thesis, especially for in intra-domain handover cases, which means handover occurs within one administrative domain. The basic idea is to take measures not only in each individual operation, but also in optimizing the overall performance of the procedure. Main measures include 1.) enhancing advertisements with QoS information in order to enable a mobile user to select the best suited access router as the handover target, 2.) separating authentication into two steps - a preliminary check serves as the first authentication to reduce the risk of DoS and check authenticity (e.g. a device's identification), 3.) integrating the binding update signaling and the QoS signaling (i.e. QoS+BU), 4.) parallelizing the QoS+BU process and the the process of authorization and authentication , and 5.) continuing immediately a mobile user's communication traffic after a successful QoS+BU process, and protecting the exchanged data with a temporary IPSec security association until the process of authentication and authorization is complete. Afterwards, a new security association with a freshly negotiated session key will replace the temporary one.

A cost-based approach is used to analyze how the security design can reduce the risks of DoS attacks. Furthermore, queueing theory is used to evaluate the efficiency of the whole protocol, comparing the existing solutions. For the sake of providing parameters to the analysis, a prototype of the proposed scheme is implemented. Also experimental analysis is carried out to evaluate the proposal.

The results of the thesis show the feasibility of the scheme in performing a re-registration procedure featuring QoS, security and mobility operations in an secure and efficient fashion. Especially, the results show its robustness against sabotage attacks with its compliance with the performance requirements. The prototypical implementation and experiments prove that the scheme is deployable in access networks in reality.

Chapter 1

Introduction

The notion of wireless mobile Internet is developing toward an integrated system of Internet and telecommunications technologies in order to fulfill the dream of human beings: *ubiquitous communication* - mobile users can move freely almost anywhere and communicate with anyone, anytime and in any form using the best service available. This demands a rapid progress in telecommunications and the Internet technologies.

One of the significant trend in telecommunications industry is the change from fixed access communications to mobile communications. The continuously exponential increase in number of mobile subscribers has been observed [42].

The scope of the Internet usage has evolved from file transfer, remote access to computers, and simple mail transfer to www-based applications. The new Internet services and applications such as email, web browsing, Internet telephony, and Internet videoconferencing makes the Internet gain great popularity.

The development in both telecommunications and Internet necessitate their integration. The telecommunication world has created various technologies such as General Packet Radio Service (GPRS) and Universal Mobile Telecommunication System (UMTS), which may be helpful in addressing the increasing demand of packet-switched data services. The evolution to an “All-IP” network is happening in the wireless networks. In the near future, every mobile devices will have an Internet Protocol (IP) address and will be connected to the Internet.

1.1 Mobile Communications

Progress in radio communications, various access technologies and coding algorithms liberates users from tethered communication devices and realizes mobile communications for them. Since the bandwidth is the most precious and scarce resource of the entire communication system, the concept of cellular structure has been introduced to achieve higher spectral efficiency. As wireless signals in one cell are attenuated after a certain distance and do not interfere, the frequency can be reused in the cells which apart sufficiently.

A seven-cell frequency reuse pattern is shown in Figure 1.1 [42]. The total radio spectrum allocated to the cellular communication services is divided into a maximum of seven sub-bands. While neighboring cells use different sub-bands, the pairs of cells that are far enough can use a same frequency sub-band. Each number shown in one cell in the figure means a separate frequency sub-band used in that cell. The cellular concept shown in 1.1 can be used in Frequency Division Multiple Access (FDMA), Time Division Multiple Access (TDMA) and Code Division Multiple Access (CDMA) systems.

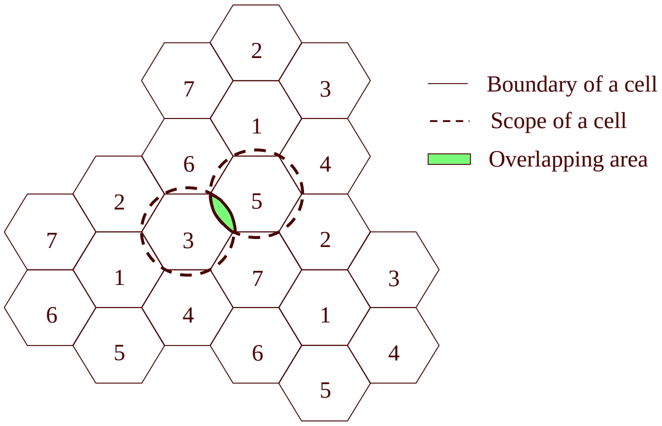


Figure 1.1: Basic Concepts of Cellular Mobile Communication and Frequency Reuse

The user of a mobile cellular network may experience two types of mobility in the network: terminal mobility and personal mobility. The first one is that a mobile device changes its network attachment point from one cell to another while an active communication session is ongoing. The process is defined as **handover**. The second one refers as personal mobility. For example in the Global System for Mobile Communication (GSM) system, a user can remove its Subscriber Identity Module (SIM) card from one terminal and insert it into another GSM-compatible terminal and still receive the same type of services from the cellular network. The thesis focuses on the first type of mobility.

Figure 1.2 illustrates an example mobile network. The cellular access network gives an overview of the architecture for the thesis. In the access network, a Gateway (GW) connects to the core network. A Mobile service Switching Center (MSC) may act as the gateway in a GSM system. The Gateway (GW) has connections to some Access Router (AR); and each Access Router (AR) links to Access Points (APs), which locate in hexagonal cells.

In general, mobility means that a user moves around and use network services with seamless handovers either within one access network or between different access net-

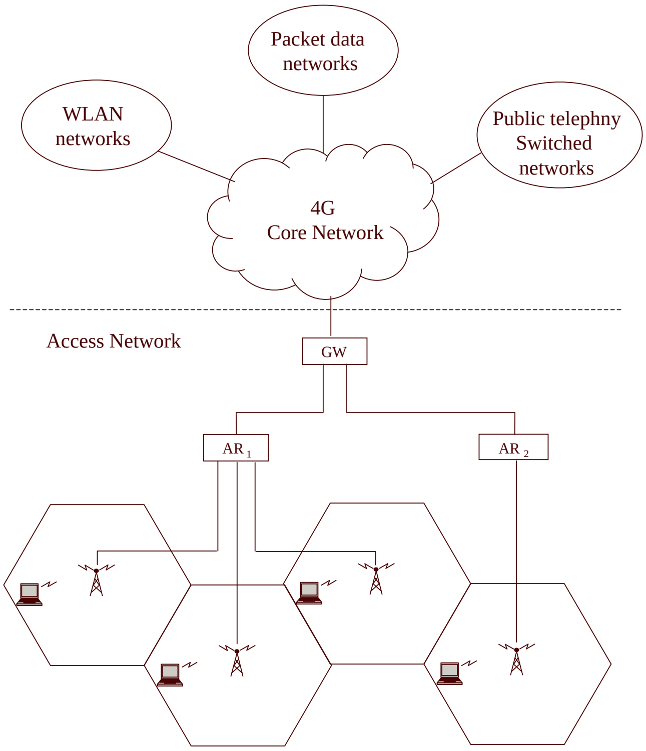


Figure 1.2: An Example Mobile Network

works. Moving from one access router to another within one access network is named as **intra-domain handover** [1]; in contrast, moving between different networks is defined as **inter-domain handover** [2].

Enabling IP-based communications on mobile devices raises new challenges to the Internet. For example, when a mobile node changes its associated access router during a session, in addition to the normal operation of the mobility management, i.e. updating the location information of the mobile user to enable the reception of IP packets at the mobile user's new IP address, two typical kinds of operations should be involved:

- **QoS**: the new path including the new access router should provide appropriate treatments to IP packets sent to/from the mobile user;
- **Security:** security checks should be performed and security associations should be established to protect the IP packets.

All the operations should be taken in a handover procedure. In such a handover procedure, the grade of the continuity of the services provided by a network is essential especially to packet loss sensitive applications and delay sensitive applications. The continuity of the services lies in two folds:

[1] it is also called micro-mobility or local movement

[2] it is also called macro-mobility or global movement

- minimal information loss during a handover;
- minimal interruption time during a handover.

The former one is important to the pack loss sensitive applications such as downloading a file, whereas the latter one is important to the real-time applications which are delay sensitive. The latter item lies in the focus of the thesis:

*the necessary operations regarding **QoS** and **security** should be performed in an **efficient** manner in a handover procedure in order to minimize the interruption to the delay sensitive applications.*

1.2 Quality of Service in Mobile Communications

Quality of Service (QoS) can be defined as a set of specific requirements for a particular service provided by a network to users. The user specifies the sorts of measures such as throughput or delay, and the network commits its bandwidth to satisfy the request.

For real-time applications such as Voice over IP (VoIP) and video conferencing, minimal delay in delivering packets is required. To minimize the delay, some measures may be taken such as reserving bandwidth along the routing path for a specific session, and scheduling packets in order to shorten the time a packet has to wait in a queue.

In a mobility scenario, QoS refers to providing the requested service even when mobile node changes its point of attachment to the network. During a handover procedure, some operations need to be performed. If performing the operations defers the delivery of real-time packets, a failure in QoS provisioning to the real-time application may result.

1.3 Security in Mobile Communications

With respect to mobile users, the signaling for QoS state establishment and mobility management in the handover procedure and user data need to be protected against various security threats such as modification of transmitted data and eavesdropping.

With respect to access networks, a service provider should use its resources efficiently. Also it should prevent any security threats such as masquerade, authorization violation and reduction of availability of resources. Therefore, before taking any actions such as updating a mobile user's location information, granting the access of resources upon a QoS request, the access network needs to perform the following operations:

- to check whether the QoS requester is the one who claims to be
  - **authentication** [3];

[3] Although authentication is not necessarily required since an authorization can also be accomplished

- to check whether the requested QoS is what the mobile host is entitled to use - **authorization**;

- if the above two checks (i.e. Authentication and Authorization (AA) checks) succeed, the access network performs following operations: updating the mobile user's location information, establishing the QoS state along a new path, enforcing a specific IP packet treatment at the new Access Router (AR), and establishing a security association to protect user data.

The AA checks themselves may cause certain security threats. For example, using strong authentication such as verifying digital signatures introduces Denial of Service (DoS) risks because this kind of authentication is relatively computationally expensive. An adversary may send a large number of requests to demand such authentications - this is one kind of DoS; the extensive bogus requests may trigger traffics for the security checks so as to degrade significantly the the signaling capacity of the access network. This is another kind of DoS.

To deal with these threats, an access network should take appropriate measures to avoid wasting resources (e.g. storage capacity, signaling capacity and computing power) on a request.

1.4 Efficient Operations in a Registration Procedure

In a handover procedure, a mobile user may not be able to receive any packet for a period of time. The interruption may result from 1.) the re-establishment of a new IP connectivity with the new AR, and 2.) the operations of enabling QoS treatment, security, and mobility management. The thesis focuses on the second factor.

The real-time applications such as VoIP demand minimal delay during a registration procedure, since on-time delivery of the packets is crucial. For example, if some VoIP packets are delayed for a period longer than the admissible end-to-end delay, even though they are stored in a buffer and are not lost, the packets may become useless. In this sense, reducing the handover latency is sometimes more important than buffering packets.

Therefore, the handover procedure featuring QoS, Security and Mobility operations should be carried out in an efficient fashion, i.e. introducing low handover latency.

The efficiency should not expose considerable security vulnerabilities. Performing security checks introduce unavoidably latency during handovers. Therefore, an access network needs to balance efficiency and the security level.

for an anonymous user [3], the authentication is assumed to be mandatory in the thesis

1.5 Thesis Formulation

Based on the above discussion, low-latency of a handover procedure, low consumption of access network's resource for the registration operations, and a reasonable level of security are crucial issues in the mobility scenarios for delay-sensitive applications.

Until recently, the protocol design activities are being carried out only for mobility support and QoS in the community. The security aspects are not considered sufficiently in the related protocol design so far.

The **goal** of the thesis is to investigate the feasibility of a secure and efficient QoS-aware mobility support in IP based cellular networks.

In the thesis, after the background information and the problem statement are described, the design space and the related activities are introduced. Based on the analysis and research, a scheme for a secure and efficient QoS-aware mobility support is proposed in the thesis. The basic idea is to take measures not only in each individual operation in a registration procedure, but also in optimizing the overall performance of the procedure.

Since this thesis does not address the IP mobility issues, Hierarchical Mobile IPv6 (HMIPv6) is selected as the mobility management protocol in order to meet the requirement of low-latency in a registration procedure. The scheme for a secure and efficient QoS-aware mobility support is illustrated in an HMIPv6 environment.

In such an environment, a handover procedure happens as the intra-domain handover case when a mobile node moves from the old AR to a new AR within an access network. Since a mobile node must have done a registration in the domain, the registration procedure is termed as re-registration, authentication as re-authentication, and authorization as re-authorization.

In the intra-domain case, a number of solutions to achieve a secure and efficient QoS-aware mobility support are proposed in the thesis:

- **Enhanced advertisements:**
  advertisements contain QoS information such as aggregate available bandwidth information. It helps Mobile Node (MN) to select the most suited access point to perform a handover.

- **A range of QoS parameters:**
  a range of QoS parameters are used rather a specific value for the resource reservation so as to avoid multiple round trips of the QoS negotiation. Also the range is used for the authorization check of a QoS request.

- **A desired value of bandwidth for the first authorization:**
  A desired value of bandwidth used for the first authorization during inter-domain handovers is unnecessarily equal to the upper bound value of the range of a QoS

parameter. If the upper bound value were used for the first authorization, when the mobile host upgraded its QoS request later, the access network had to re-perform the authorization involving the mobile host's home domain unavoidably. Therefore, performing the first authorization with a desired value (which is independent to the upper bound value) can not only solve the problem, but also avoid distributing the subscribed value of bandwidth of a mobile host from its home domain to the visited domain.

- **Two-step authentication:**
  when receiving a QoS request, an access router performs a preliminary authentication check as the first step of authentication. If the verification is successful, the access network regards the mobile user as a credible user and thus commits its resources to process its request. During processing the request, the access network checks the authenticity of the mobile user as the second step of authentication.

- **Integration of the QoS signaling and the mobility signaling:**
  the information for registering new IP address (the registration is called Binding Update (BU) process) is piggybacked in the QoS signaling (termed as QoS+BU or BU+QoS). The local binding update is operated only when the the QoS can be satisfied in the new path.

- **Parallelizing the QoS+BU process and the re-authorization process:**
  AR initiates the QoS+BU process and the re-authorization process at the same time after the first-step authentication succeeds.

- **A temporary IPSec Security Association (SA)**:
  since the QoS+BU process and the re-authorization process happen in parallel, the results of the two processes may arrive at the AR at different times. When QoS+BU process arrives earlier, a temporary IPSec SA is established between the MN and the AR to secure the user data before the mobile user gets the acknowledgement of the re-authorization process.

In the protocol specification, the operations of each network node during an MN's intra-domain handovers and session updates are illustrated.

To analyze the involved security protocol, a cost-based approach is used. To evaluate the performance of the cookie mechanism in the intra-domain handover case, an analysis of DoS is performed comparing to a re-registration procedure which uses an AAA protocol for security checks.

To complete a re-registration procedure featuring QoS, security and mobility operations, the state-of-art solutions for QoS and security aiming at the seamless handover support are studied to analyze how they can work together. A comparison between the proposal in the thesis and the possible combinations of the solutions is made in terms of *mean response time*, *packet losses*, and *cpu computing load*.

A prototype of the formulated proposal in the thesis is implemented. Measurements are taken in order to evaluate the proposal and provide parameters to the mathematical analysis.

1.6 Thesis Outline

The remainder of the thesis is organized as follows:

In Chapter 2, the background information is introduced in the aspects of IP mobility support, QoS, authentication, authorization and Denial of Service (DoS). After a general introduction of IP mobility support, the chapter mainly addresses on QoS and security issues. In security, Denial of Service (DoS) is focused in the context of QoS.

In Chapter 3, the problems which the thesis addresses are discussed in detail. The specific problems are identified in the defined scope of the thesis: to achieve a secure and efficient QoS-aware IP mobility support.

To address the identified problems, the general design space is discussed, and some related work is introduced in Chapter 4.

Based on the design outline specified in Chapter 4, the formulated proposal is described in Chapter 5. Also the specification of the proposal is given in the chapter.

In order to analyze the performance of the proposal, a couple of approaches are used: a cost-based approach is used to analyze the formulated security protocols (see Chapter 6); a mathematical analysis is carried out to evaluate the performance of the security protocols in a complete re-registration procedure (see in Chapter 7); a mathematical analysis is performed to evaluate the overall performance of the proposal, comparing with the possible schemes for the re-registration procedure (see Chapter 8); an experimental evaluation is done to justify the proposal and provide parameters to the above-mentioned mathematical analysis. The experimental environment and evaluation results are presented in Chapter 9.

Finally, conclusions and outlook are discussed in Chapter 10.

Chapter 2

Background Information

Since the thesis focuses on the QoS provisioning and security issues in the IP mobility environment, this chapter introduces the fundamentals of the corresponding topics including

- IP mobility,
- Quality of Service (QoS),
- authentication,
- data protection over wireless channel: confidentiality and integrity,
- authorization, and
- Denial of Service (DoS).

2.1 Mobility Support in IP based Network

2.1.1 General Overview

A communication network exchanges information between users. It provides either a connection-oriented or a connection-less service to applications. For a connection-oriented service, applications executed in the devices exchange control messages to create proper contexts along a path before sending messages with user data. For a connection-less service, no handshake procedure prior to transmission of messages is needed. Applications simply send the messages. In the last decades, a communication device was typically tethered to a site. With the technological progress, a telephone device can be mobile and a computer becomes portable or hand-held.

Wireless networks use the radio spectrum to connect a mobile terminal to an access point or base station. The access point or base station is connected to an edge router, namely **access router** of an IP based wireless access network. Since this thesis addresses IP based access networks, the evolution of IP based networks will be first introduced briefly, then the mobility problems of the IP network and the corresponding solutions will be discussed.

2.1.2 Evolution of IP Based Networks: from IPv4 to IPv6

In the early 1990s, the Internet Engineering Task Force began an effort to develop a successor to the IPv4 protocol. A prime motivation for this effort was the realization that the 32-bit IP address space was beginning to be used up. Since more and more IP nodes are attached to the Internet, and IP addresses are being allocated at a great rate. To meet this need for a large IP address space, a new IP protocol - IPv6, was developed [54]. The IETF specifications for IPv6 contain a lot of information about the transition from IPv4 to IPv6 [22].

IPv6 has the following main features:

- **bigger address space:** a IPv6 address has 128-bit [38]. With the enlarged address space, every IP-based devices or machines can have at least one globally unique IP address.
- **mobility:** Mobile IP is one of the requirements for any IPv6 stack. Mobile IPv6 solved the problem of Mobile IPv4 that establishs a reverse tunneling to a mobile node's home agent [63]. See more detailed discussion in the following section.
- **security:** IPv6 protocol stack is needed to include IP Security (IPSec) [47, 96]. IPSec allows authentication, encryption, and compression of IP traffic. The network layer security protocol provides the security services independent of applications.

2.1.3 The General Mobility Problems in IP Based Networks

Originally, IP networks have been designed under the assumption that hosts are stationary. This assumption implies that the IP address does not change and a host is reachable from other hosts by an IP address that does not change. Data is carried by means of IP packets which contain source and destination addresses. Internet routers inspect the destination address contained in an IP packet. They make a forwarding decision based on the network part of the IP destination address and forward packets to the determined next hop. Consequently, this addressing scheme puts restrictions on the address usage. In particular, an IP address can only be used within the network of its definition [26].

As shown in Figure 2.1, if a mobile host moves to a new network, the old IP address becomes topologically incorrect. Therefore, a new topologically correct IP address must

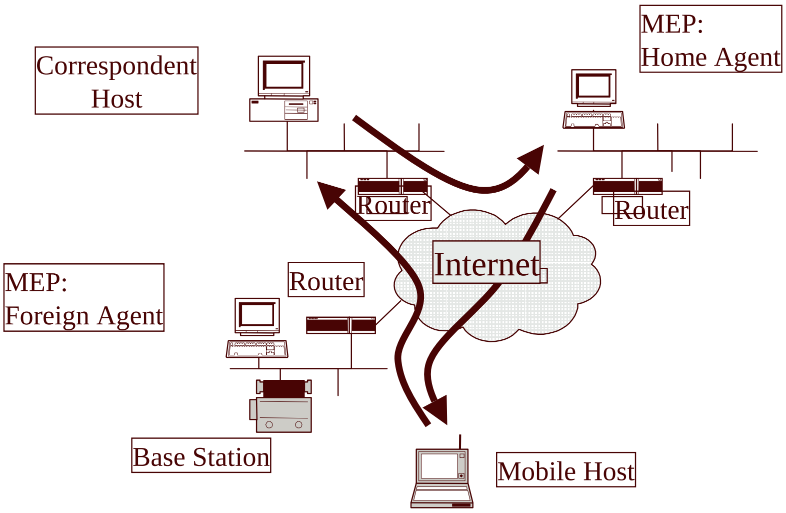


Figure 2.1: The General IP Mobility Problem [26]

be assigned to a mobile host. In the TCP/IP protocol suite, applications access communication services through a socket layer - a protocol independent interface to the protocol-dependent below. When an application establishes a session between two hosts, the IP address of the application‘s source node is used as the source address for the session. Meanwhile, the IP address is interpreted as a host identifier.

When a mobile host moves to a new network then the network part of the mobile host’s IP address does no longer match the IP network address of the new point of attachment. The same problem occurs if the network is divided into subnetworks: When a mobile host moves to a new subnetwork the subnet-id becomes incorrect. The assignment of a new IP address, which is topologically correct, enforces the closure and re-opening of existing communication sockets. In fact, sockets are bound to source and destination addresses. Hence, the re-establishment pertains to the mobile as well as to a correspondent host communicating with the mobile host and disrupts communication service. That an IP address represents both host identification and location - is the fundamental mobility problem in IP based networks. This problem needs to be solved by mobility concepts.

In summary, the identified problems are as follows [26]:

- current IP based protocols assume to be stationary hosts;
- IP address reflects both identity and location;
- when hosts become mobile, the ongoing network sessions are disrupted;
- No "user mobility"-concept in Internet as it is common in e.g. GSM.

2.1.4 Mobile IP: the Solution for Mobility Support in IP Networks

Host-specific routing has severe scalability, robustness, and security concerns. Changing a node's IP address as it moves makes it impossible for the node to maintain any ongoing communications as it changes links.

Mobile IP is thus a solution for mobility on the global Internet which is scalable, robust, secure, and which allows nodes to maintain all ongoing communications while changing links. Specifically, Mobile IP provides a mechanism for routing IP packets to mobile nodes which may be connected to any link while using their permanent IP address.

Mobile IPv4 [75] is a mobility management protocol in the IPv4 Internet, whereas Mobile IPv6 [43] is a protocol which allows Mobile Node (MN) to remain reachable while it moves around in the IPv6 Internet. Since the work in the thesis is based on Mobile IPv6 (MIPv6), Mobile IPv6 is introduced in more detail. The operations of Mobile IPv6 are summarized as follows [92]:

- a mobile node determines its current location using the IPv6 version of Router Discovery;
- when the mobile node connects its home link, it acts like any fixed host; otherwise, it uses IPv6-defined address auto-configuration to acquire a (col-located) care-of address on the foreign link;
- the mobile node notifies its home agent and selected correspondent nodes of its Care-of Address (CoA) provided it can do so with security consideration;
- packets sent by correspondent nodes can either be routed to the mobile node's home network, where the home agent tunnels them to the care-of address; or be sent directly to mobile node using an IPv6 Routing Header, which specifies the mobile node's care-of address as an intermediate destination;
- in the reverse direction, packets sent by a mobile node are routed directly to their destination using no special mechanisms; however, in the presence of ingress filtering, the mobile node can tunnel packets to its home agent using its care-of address as the IP Source Address of the tunnel.

Figure 2.2 shows the basic operation of Mobile IPv6.

2.1.5 Global Mobility and Local Mobility

In this thesis, the terms of global mobility, macro mobility and inter-domain handover are interchangeable. Local mobility, micro mobility and intra-domain handover are also interchangeable.

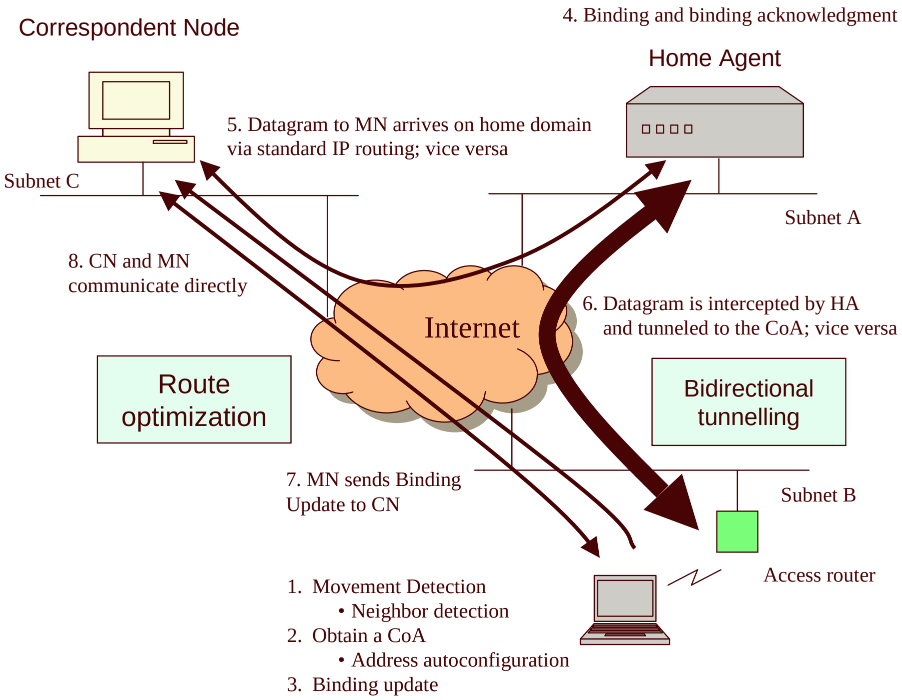

Correspondent Node
4. Binding and binding acknowledgment
Home Agent
5. Datagram to MN arrives on home domain via standard IP routing; vice versa
Subnet C
Subnet A
8. CN and MN communicate directly
Internet
6. Datagram is intercepted by HA and tunneled to the CoA; vice versa
Route optimization
Bidirectional tunnelling
7. MN sends Binding Update to CN
Subnet B
Access router
1. Movement Detection
• Neighbor detection
2. Obtain a CoA
• Address autoconfiguration
3. Binding update

Figure 2.2: Basic Operation of Mobile IPv6

As the name suggests, **inter-domain handover** means that a mobile node roams from one administrative domain to another. Inter-administrative domain mobility requires global mobility management.

Intra-domain handover occurs when a mobile node roams within an administrative domain. In such a case, the Mobile IP solution is not optimal to support micro mobility. Firstly, the mobile node needs to generate significant signaling to its home agent and the corresponding node for a local movement. Secondly, the procedure creates a considerable delay for the location update. Therefore, it may cause communication interruption and packet loss during a handover. To overcome the problems, two major approaches are introduced: Proxy Agents Architecture Schemes (PAA) and Localized Enhanced-Routing Schemes (LERS) [58].

- PAA: These schemes extend the idea of Mobile IP into a hierarchy of Mobility Agents. Examples include and Hierarchical Mobile IPv6 (HMIPv6) [91].

- LERS: These schemes introduce a new, dynamic Layer 3 routing protocol in a localized area. The examples include HAWAII [77], Cellular IPv6 [88].

More detailed discussions on micro-mobility management protocols are available in [26].

2.2 Quality of Service

QoS can be defined as a set of specific requirements for particular services provided by a network to users. These requirements are usually described by using some quantitative figures such as connection speed or delay. The services can be specified so that particular applications (e.g. voice, video, streaming audio) can stratify the perception requirements of human beings.

The current global Internet service is based on the so-called best-effort service. This service does not guarantee anything, even delivering the IP packets within the network. Once a packet is generated and sent to the Internet for delivery to a destination host, the network does not guarantee any specified delivery time (delay), the speed at which the packet will be forwarded (data rate and throughput), the available bandwidth for delivering the packet, or even that the packet does not get lost during this delivery.

In recent years, the issues to provide guaranteed QoS to applications have been posed in designing the next-generation telecommunications. In such services, particular treatments must be guaranteed on IP packets in the metrics of e.g. delay, average, minimum, or peak throughput, bandwidth, or loss probability.

There are four fundamental principles in providing QoS in data and IP networks [42].

- **Packet classification:** Packet classification enables an application to receive a specific treatment at a router. For example, if a router needs to provide 1.5 Mbps to application A and 0.5 Mbps to application B, it should be able to classify and mark different packets at the input port of one AR so that the router can share the total capacity proportionally to the two applications. Both IP version 4 and 6 have the corresponding field in IP header to enable packet classification. Thus, the different types of packets generated from the applications can be differentiated by the router.

- **Packet Isolation:** In order to prevent the misbehavior of packet classification done by individual applications, there should be some entity within the network to monitor the behavior of applications and their use of the network resources. Therefore, in addition to packet classification, a monitoring and controlling scheme is needed to ensure that no one uses more resources than what they have been allocated. Such a scheme usually also provides for distributed systems such as reliability and minimal exchange of control messaging.

- **Efficient Resource Management:** Resource management is an important issue to avoid the waste of resources when resource is allocated to an application but it is not used by the application for a period of time. One way to manage resources is to manage queues. For example, if a router partitions its resources into two parts of 1Mbps and 0.5Mbps and allocates them to applications A and B, it can maintain two queues for allocations. Maintaining the two queues can be done by by a software easily. Thus, the allocations can be managed and reconfigured dynamically, avoiding the waste of resources.

- **Traffic Load Control:** When a network has limited capacity and it has to share the resources among applications, the network needs to enforce a policy to allocate the limited resources to applications. For example, the resources can be allocated on a first-in-first-served basis, or on a user subscription priority basis, or other policies.

In general, two different approaches have been proposed: Integrated Services(IntServ) [14] in 1994 and Differentiated Services (DiffServ) [12, 7, 69] in 1998.

2.2.1 IntServ

IntServ [101] has been introduced in IP networks in order to provide guaranteed and controlled services in addition to the already available best-effort service. Each traffic flow in this service can be classified under one of the three service classes:

- Guaranteed-service class [90]: it provides for delay-bound service agreements such as voice and other real-time applications, which require severe delay constraints.

- Controlled-load service class [100]: it provides for a form of statistical delay service agreement, for example, with a nominal mean delay.

- Best-effort service: it has been included to match the current IP service mainly for interactive burst traffic (e.g. web), interactive bulk traffic (e.g. FTP), and background or asynchronous traffic (e.g. email).

A resource reservation protocol (RSVP) [102] is used for the guaranteed and controlled-load services, which are based on quantitative service requirements and require signaling and admission control in network nodes. RSVP is a signaling protocol used to reserve resources in the routers, in a hop-by-hop basis, considering the applications requirements for a given IP flow.

Figure 2.3 illustrates an example of call setup process using IntServ [54].

The main advantages of the IntServ are that it provides service classes that closely match different application requirements; it leaves the existing best-effort service unchanged and the forwarding mechanism in the network unchanged.

On the negative side, the architecture of IntServ requires that all involved nodes along an end-to-end path need to support the service agreement for a given flow in order to guarantee an end-to-end service.

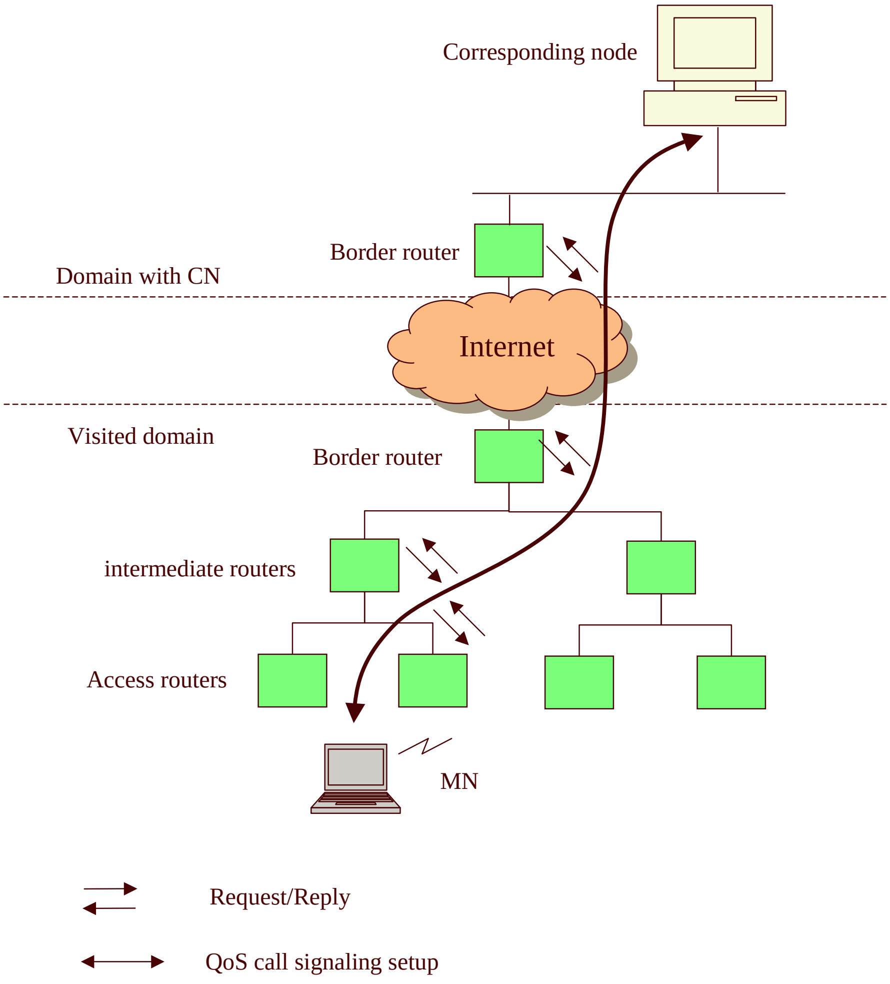


Figure 2.3: An Example of Call Setup Process using IntServ

The IntServ and RSVP proposals have failed to become an actual end-to-end QoS solution, mostly because of the scaling problems in large networks and the need to implement RSVP in all network elements from the source to the destination.

2.2.2 DiffServ

DiffServ came to remedy the disadvantages of IntServ in providing QoS in IP networks. It aims at providing simple, scalable, and flexible service differentiation using a hierarchical model. The resource management divides into two domains:

- Inter-domain resource management: unidirectional service levels are agreed at each boundary point between a customer and a provider for traffic entering the provider network.

- Intra-domain resource management: the provider is solely responsible for configuration and provisioning of resources within a domain.

DiffServ allows the network to differentiate traffic streams, using different Per-Hop-Behaviors (PHB) when forwarding the IP packets of each stream [35, 21]. The advantage of such a scheme is its scalability since many IP flows can be aggregated in the same traffic stream or behavior aggregate (BA). The PHB applies to an aggregate and is characterized by a DiffServ Code Point (DSCP) marked in the header of each IP packet [42].

Figure 2.4 illustrates DiffServ services network architecture and the three types of DS routers.

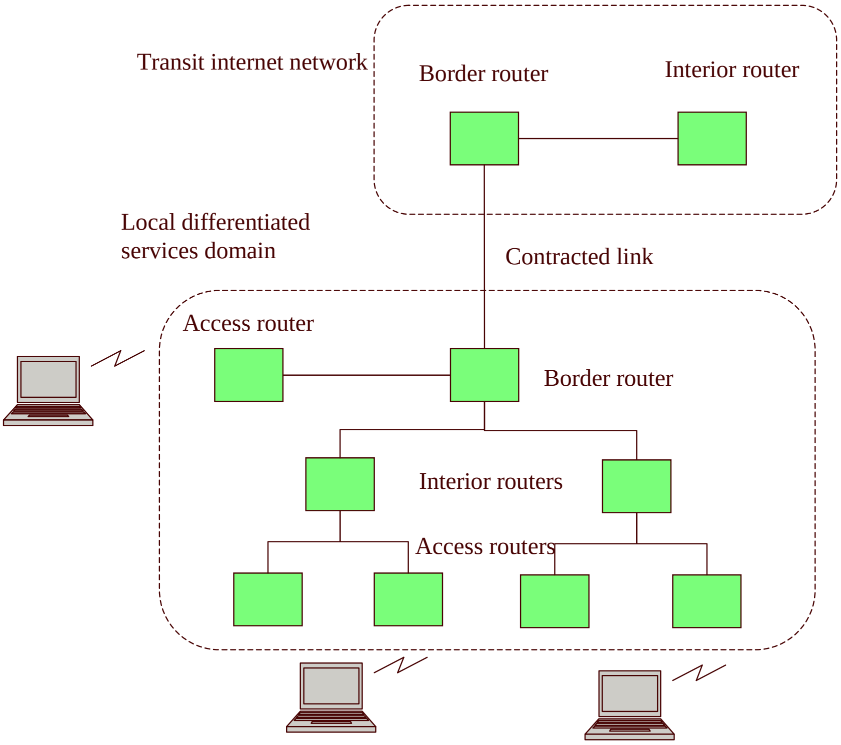


Figure 2.4: A DiffServ Services Network Architecture

The difference from IntServ in which all control and resource management are performed on an end-to-end basis, is that in DiffServ the local network has to share the resources allocated by the outside network or service provider to its users. Scalability, simplicity, and flexibility of DiffServ come from this hierarchical management.

In the DiffServ architecture, a Service Level Agreement (SLA) is provided to govern the traffic handling between a local network and the service provider network. The local network provides the required services to its users. Per flow state is also avoided in DiffServ since individual flows are aggregated in classes and will be supported by the local network resource management using the available resources provided on the basis of the SLA.

The SLA could be static (negotiated and agreed on a long-term basis) or could be dynamic, which changes more frequently. The local network is then responsible for providing DiffServ to end users within the network. This is usually done by marking packets with specific flags shown in TOS field or IPv4 or the TC field of the IPv6.

DiffServ comes with some advantages and disadvantages. DiffServ provides the kind of discrimination based on payment for service. Traffic classes are accessible with additional signaling as a traffic class is a predefined aggregate of traffics. Network management will be simpler in DiffServ compared to IntServ, since the classification for the traffic needs to be performed at the end systems.

On the other side, DiffServ tries to keep the operating mode of the network simple by pushing as much complexity as possible onto the network provisioning and configuration. DiffServ also does not make the provision of several services with different qualities within the same network easier.

With the comparison of IntServ and DiffServ, IntServ requires flow-specific state for each each flow at the routers. State information will be increased in accordance with the number of flows; in contrast, DiffServ is simpler and more scalable. The reason for the scalability of DiffServ is that the per-flow service is now replaced with per-aggregate service. The complex processing is also now moved from the core network to the edge.

2.2.3 IntServ over DiffServ

Recently, it has been proposed to apply the IntServ end-to-end model across a network containing one or more DiffServ regions. This approach has some advantages because it removes the per-flow processing from the core routers. However, the per-flow processing remains essential at both the edge and border routers [8].

The basic requirements and assumptions are that the resource signaling is done with RSVP and that a mapping at the border routers for RSVP-based reservations to DSCP values. Routers within the DiffServ region may be able to produce RSVP messages, even though the forwarding operation is purely based on the DSCP values. This allows more accurate resource coordination within the DiffServ domain. Also a SLA between non-DiffServ and DiffServ regions is needed.

The primary benefit of combining IntServ and DiffServ is the increased scalability, provided through the aggregate traffic control of DiffServ.

2.2.4 QoS Support in Mobility

During a handover procedure, traffic flows should obtain QoS treatment as soon as a new path has been set up. The QoS signaling should only add a minimal latency to the (re-)registration procedure.

So far the protocols for supporting mobility and QoS have been worked on separately. From the QoS point of view, the problems with mobility in a wireless access network and mobility-related routing schemes are related to providing the requested service when the mobile node changes its point of attachment from one access router to another. When the service assured to the mobile node can not be provided after a handover, a violation of the assured QoS may occur. In order to avoid such a violation, the mobile node may perform a handover only when the requested service is assured in the new path [29].

In macro mobility, although MIPv6 solution meets the goals of operational transparency and handover support, it is not optimized for managing seamless mobility in large cellular networks in terms of minimal delay. The latency involved in communicating the update messages with a corresponding node which locates remotely do harm to QoS state establishment especially for real-time applications.

In micro mobility, since IntServ stores per-flow state in each router, thus the movement triggers “local repair” of routing and resource reservation within the network [85]. For DiffServ, even though no state needs to be updated, the offered service level may vary. Both IntServ and DiffServ have been extended to cope with tunneling and changes to the IP address [13]. Some coupling of the macro and micro mobility protocols and the QoS architecture may be necessary [30, 89].

2.3 Authentication and Cryptographic Algorithms

2.3.1 An Overview of Authentication

Authentication is defined as the act of verifying a claimed identity, in the form of a pre-existing label from a mutually known name space, as the originator of a message (message authentication) or as the end-point of a channel (entity authentication).

Authentication is the most fundamental security service as all other security services build upon it. The following two principle variants of authentication are distinguished [93, 83]:

- **Data origin authentication** is the security service that enables entities to verify that a message has been originated by a particular entity and that it has not been altered afterwards. A synonym for this service is **Data Integrity**. The relation of data integrity to cryptographic protocols is two fold:

Table 2.1: Cryptographic Algorithms

	Typical Cryptographic Algorithms
Symmetric Cryptography	DES, AES, CR4
Asymmetric Cryptography	RSA, Diffie-Hellman Key Exchange, EIGamal
Integrity Check Values	MD5, SHA-1, HMAC

- There are cryptographic protocols to ensure data integrity.
- Data integrity of messages exchanged is often an important property in cryptographic protocols, so data integrity is a building block to cryptographic protocols.

- **Entity authentication** is the security service, that enables communication partners to verify the identity of their peer entities. In principle, it can be accomplished by various means:
  - Knowledge: e.g. passwords
  - Possession: e.g. physical keys, cards or tokens
  - Immutable characteristic: e.g. biometric properties like fingerprint and retinal pattern
  - Location: e.g. evidence is presented that an entity is at a specific place. (For example, people check rarely the authenticity of agents in a bank)
  - Delegation of authenticity: the verifying entity accepts, that somebody who is trusted has already established authentication

As in communication networks, direct verification of the above means is difficult or insecure. Therefore, cryptographic protocols have been developed for the purpose of authentication, confidentiality and integrity in the communication network.

There are two main categories of cryptographic algorithms which serve as fundamental building blocks of authentication protocols: Encryption algorithms and Integrity check values. Encryption algorithms may be divided into symmetric encryption algorithms and asymmetric encryption algorithms; Integrity check values may be further divided into Modification Detection Code (MDC) and Message Authentication Code (MAC). An overview of some common cryptographic algorithms are shown in Table 2.1 .

2.3.2 Symmetric Cryptography

Symmetric Cryptography uses the same key $K_{A,B}$ for encrypting and decrypting a message between Nodes A and B.

In the symmetric block cipher algorithms such as Data Encryption Standard (DES) [65] and Advanced Encryption Standard (AES) [66], a plaintext P is first segmented into blocks $p_1, p_2, ...$ with the block size of b. Then the plaintext is encrypted with a key K, which has the size of j and $j \leq b$. The encryption results denote as $c_1, c_2, ...$ The final ciphertext c is the combination of them.

In 1973 the National of Standards (NBS, new National Institute of Standards and Technology, NIST) requested for proposals for a national cipher standard. The proposal from IBM was adopted in 1977 as DES. From then on, DES was widely used especially in financial applications.

The DES algorithm is a recirculating, 64-bit block product cipher whose security is based on a secret key. DES keys are 64-bit binary vectors consisting of 56 independent information bits and eight parity bits. (Since 56 independent bits are used in a DES key, 256 such tests are required to guarantee finding the secret to a particular key. The expected number of tests to recover the correct key is 255 [84].

When DES was reaffirmed by NBS in 1993, ending December 1998, the following statement was included in the standard:

"At the next review (1998), the algorithm specified in this standard will be over twenty years old. NIST will consider alternatives that offer a higher level of security. One of these alternatives may be proposed as a replacement standard at the 1998 review."

As the knowledge of cryptanalysis developed and the need for fast software encryption and decryption rose, advances to the DES were required.

The official announcement of the AES standard took place in 2001. The lengths of key and block can be 128, 192 or 256 bits. In case of the 128 bits as the sizes of block and key, the algorithm operates on the fashion of a $4x4$ array for state and key. 10 rounds are taken for ciphering. Round 1 to 9 has the four different operations:

- ByteSub: a non-linear byte substitution;

- ShiftRow: the rows of the state array are cyclically shifted by various offsets;

- MixColumn: the columns of the state array are considered a sploynomials over $GF(2^8)$ and multiplied modulo $x^4 + 1$ with a fixed ployomial $c(x)$, given by $c(x) = \text{"03"}x^3 + \text{"01"}x^2 + \text{"01"}x + \text{"02"}$;

- RoundKey: a round-key is XORed with the state array;

whereas Round 10 does not have the MixColumn operation.

In the symmetric stream cipher algorithms such as RC4, one bit or one byte in a digital data stream is encrypted. In RC4, the plaintext P is XORed with a pseudo-random sequence $RC4(IV, K)$, where K is a key and IV is an initialization vector.

The IEEE802.11 standard defined the Wired Equivalent Privacy (WEP) encapsulation of 802.11 data frames to provide data privacy to the level of a wired network. But the 802.11 design community conceded that the WEP encapsulation failed to meet the design goal because WEPs use of 40-bit RC4 as is encryption mechanism. Even though it was suggested to migrate from 40-bit to 128-bit RC4 keys, the the WEP encapsulation was proved to be unsafe [99].

In summary, the symmetric cipher algorithms are mainly used for the security services of confidentiality. The symmetric cryptography is adapted in the design of IPSec SA establishment in the thesis.

2.3.3 Asymmetric Cryptography

Asymmetric cryptography uses two different keys $-K$ and $+K$ are used for encryption and decryption, where $-K$ denotes the private key being known only by the entity A; $+K$ denotes the public key being known to public [93].

Applications of asymmetric cryptography include confidentiality, authentication and non-repudiation. Confidentiality relies on the fact that if B encrypts a message with A's public key $+K_A$, he can be sure that only A can decrypt it using $-K_A$; the last two relates to signing: if A encrypts a message with his own private key $-K_A$, everyone can verify the signature by decrypting it with A's public key $+K_A$; also A can not deny a transmitted message since only A knows its private key and is able to generate the signature.

The security of the asymmetric cryptography algorithms depends on the difficulty of certain mathematical problems (e.g. RSA is based on the difficulty of factoring, whereas Diffie-Hellman and EIGamal based on the difficulty of computing discrete logarithms). In practice, asymmetric cryptographic operations are significantly slower than the symmetric operations.

Since the verification of a digital signature is relatively resource and time consuming due to the computational burden, using asymmetric cryptography for authentication is not favorable in the low-latency handover procedure.

2.3.4 Integrity Check Value

As the name suggests, Integrity Check Values are used to check the integrity of messages: whether a message has been modified during its transmission. Moreover, it can serve as an authenticator to enable a receiver to verify the authenticity of a message.

In addition to **Message Encryption** which means the ciphertext of the entire message serves as its authenticator, there are two other classes which are categorized as "Integrity Check Value":

- **Hash Function**: A public function that map a message of any length into a fix-length value that serves as the authenticator;
- **Message Authentication Code (MAC)**: A public function of the message and a secret key that produces a fixed-length value that serves as the authenticator.

Cryptographic Hash Functions

A hash value is generated by a function H of the form

$$h = H(M)$$

where M is a variable-length message and $H(M)$ is the fixed-length hash value. The hash value serves as a "fingerprint" of a message. A hash function H has the following important properties [93]:

- **one-way property**: for any given code h, it is computationally infeasible to find x such that $H(x) = h$;
- **weak collision resistance**: for any given block x, it is computationally infeasible to find $y \neq x$ with $H(y) = H(x)$;
- **strong collision resistance**: it is computationally infeasible to find any pair (x, y) such that $H(x) = H(y)$.

These properties are important for the design of the cookie mechanism (see the security analysis chapter).

MAC

It is assumed that two communicating peers A and B share a secret key K. A calculates a MAC as a function of a message and the key

$$MAC = C_k(M)$$

The MAC is appended to the message and sent to B. B can verify whether the message is sent from A by computing a MAC of the received message and the K. If the computation result matches the MAC appended with message, the verification is successful. Thus A is authenticated by B with the evidence that A knows the key k.

It is required that if an attacker observes the message M and the MAC $C_k(M)$, it should be computationally infeasible for the attacker to construct a message M' such that $C_k(M') = C_k(M)$.

This property, also called **computation-resistance**, implies the property of **key non-recovery**, that is the key K can not be recovered form the message and its MAC.

These properties are also important for the design of the cookie mechanism (see the security analysis chapter).

Table 2.2: Denotations in HMAC Structure [93]

H	embedded hash function (SHA-1 is used in the cookie mechanism design in the thesis)
M	message input to HMAC
Y_i	*i*th block of M, $0 \leq i \leq L-1$
L	number of blocks in M
b	number of bits in a block
n	length of hash code produced by embedded hash function
K	secret key
K^+	K padded with zeros on the left so that the result is b bits in length
ipad	00110110 repeated b/8 times
opad	01011010 repeated b/8 times
IV	initial value with the length of n bits

HMAC

In addition to use a symmetric block cipher to produce a MAC, a cryptographic hash code can be used for the purpose with the feature that cryptographic hash functions such as MD5 and SHA-1 generally execute faster in software than symmetric block ciphers such as DES.

This property matches the requirement to design a cookie mechanism (see the design and specification chapter). Since HMAC-SHA1 is used in the design of the cookie mechanism which serves as the first authentication, the HMAC algorithm is introduced more detailed as follows.

The overall operation of HMAC is illustrated in Figure 2.5. The corresponding denotations are shown in Table 4.1.

The $HMAC_K$ is produced as follows:

1. First the key K is padded to length of b bits by appending zeros at its end, the result is K^+;
2. then K^+ is XORed with $ipad$ to generate a $b-bit$ block S_i;
3. an $n-bit$ hash code is computed by applying the hash function to the stream which consists of S_i and the message M;
4. K^+ is XORed with $opad$ to generate a $b-bit$ block S_0;
5. another hash code is calculated by applying the hash function to the stream consisting of S_0 and the result of Step 3.

The HMAC can be expressed as

$$HMAC_K = H[(K^+ \oplus opad)||H[(K^+ \oplus ipad)||M]]$$

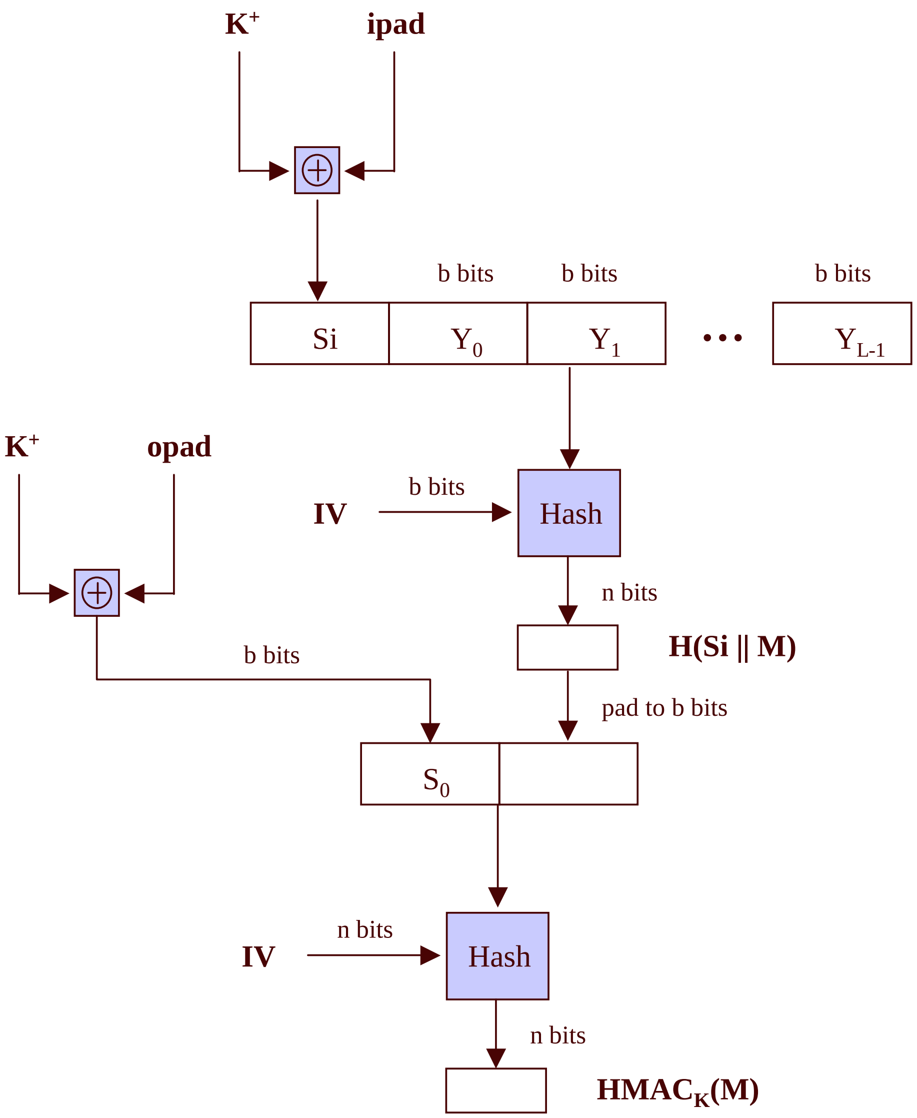


Figure 2.5: HMAC Structure [93]

2.4 Authentication Applications

2.4.1 Kerberos

Kerberos [61, 50] is an authentication and access control service for workstation clusters that was designed at Massachusetts Institute of Technology (MIT) in 1980s. It is a prominent example of arbitrated authentication which means that two or more entities make use of a so-called Trusted Third Party (TTP) to verify the authenticity of one another.

Kerberos provides the following services:

- **Authentication:** A will authenticate to an **authentication server (AS)** which will provide a “ticket” - **ticket-granting ticket** to demand access for services;

- **Access Control:** A provides the "ticket" to a **ticket-granting server (TGS)** to grant access to a service provided by server *S1*;
- **Key Exchange: AS** provides a session key for the communication between A and **TGS**; **TGS** provides a session key for the communication between A and **S1**.The use of these session keys also serves for authentication purposes.

2.4.2 X.509

X.509 [41] defines a framework for provision of authentication services by means of certificates. It is a direct authentication: the authentication is handled without direct involvement of a trusted third party. In such a case, some security infrastructure is needed to enable them to establish trust-relationship.

2.4.3 Authentication in IP Mobility Environment

When reasoning about and designing an authentication infrastructure and protocol for Mobile IP, the following authentication relations should be considered [83]:

- Authentication between the mobile node and its home network serves to counter hijacking attacks, by which an attacker may obtain access to the IP packets destined to the mobile node.
- Authentication between the mobile node and the visited network serves to be able to control access to network resources and to enable secure accounting of network resource usage.
- Authentication between the visited network and the home network also serves to control which mobile node may use network resources and ensure accounting of the resource usage. Additionally it allows to control which networks may be accessed by a mobile node.

In the security aspects of Mobile IPv6, the mobility registration messages must always be authenticated. The same procedure is to be used when updating location information to any correspondent node or router. Also the mobile node must be able to identify itself when using its care-of address as a source address as sending packets to correspondent nodes. The home address which is previously known to the correspondent node can be used as an identifier.

Figure 2.6 shows the entities involved in a Mobile IP registration supported by an infrastructure. The joined AAA and Mobile IP authentication procedure assumes some static trust relationships that are depicted with continuous lines in Figure 2.7. By making use of these static trust relationships, the AAA and Mobile IP registration procedure allows to create dynamic trust relationships which are depicted by dotted lines in Figure 2.7.

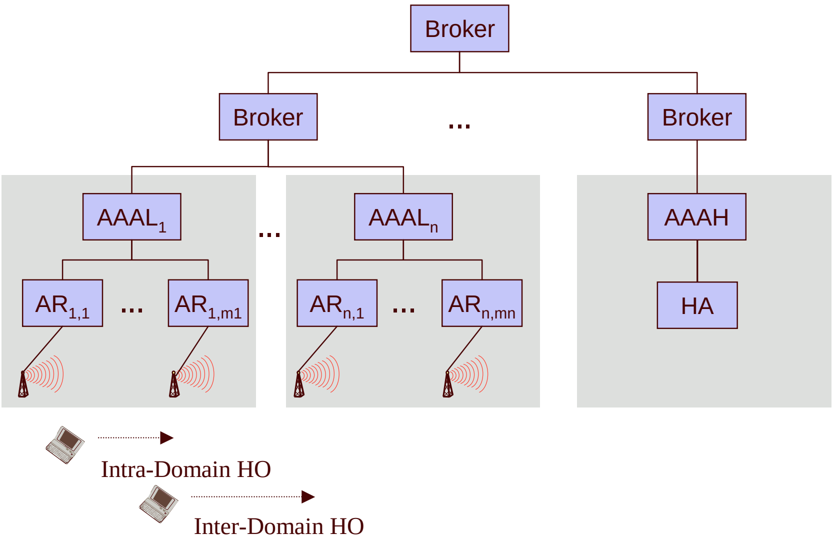


Figure 2.6: Entities Involved in Integrated AAA and Mobile IP Authentication [83]

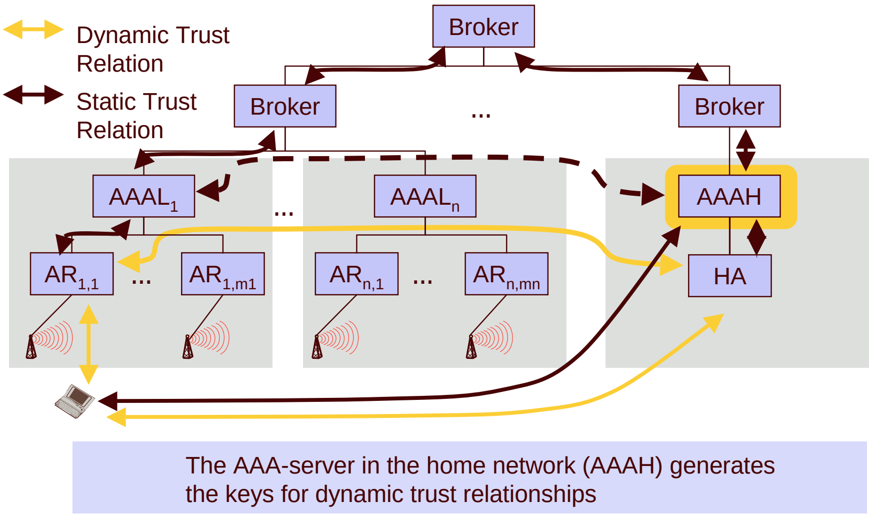


Figure 2.7: Trust Relationships in AAA and Mobile IP Joint Architecture [83]

2.5 Data Protection Over Wireless Channel

In the IEEE 802.11 Wireless LAN (WLAN) environment, Wired Equivalent Privacy (WEP) claims to provide the security services of data origin authentication, confidentiality and data integrity at link layer. However, none of the claims holds. One of the recommended solution is Virtual Private Network (VPN) enabled by IP Security (IPSec).

In IP based cellular networks, IPSec can also be used to provide the security services of data origin authentication, confidentiality and data integrity at the Network layer. Since IPSec is applied in the design of data protection over wireless channel, IPSec is introduced in more detail in the following.

2.5.1 An Overview of IPSec

IP Security (IPsec), which is an open, standard security technology, is developed by the Internet Engineering Task Force (IETF). IPsec provides cryptography-based protection of data at the IP layer of the communications stack. No changes are needed for existing applications. IPsec is the industry-standard network-security framework chosen by the IETF for both the IPv4 and IPv6.

IPsec protects signaling and data traffic using the following cryptographic techniques:

- **Authentication:** Process by which the identity of a host or end point is verified.
- **Integrity Checking:** Process of ensuring that no modifications were made to the data while in transit across the network.
- **Encryption:** Process of ensuring privacy by "hiding" data and private IP addresses while in transit across the network.

Authentication algorithms prove the identity of the sender and data integrity by using a cryptographic hash function to process a packet of data (with the fixed IP header fields included) using a secret key to produce a unique digest. On the receiver side, the data is processed using the same function and key. If either the data has been altered or the sender key is not valid, the datagram is discarded.

Encryption uses a cryptographic algorithm to modify and randomize the data and key to produce encrypted data known as ciphertext. Encryption makes the data unreadable while in transit. After it is received, the data is recovered using the same algorithm and key (with symmetric encryption algorithms). Encryption must occur with authentication to verify the data integrity of the encrypted data.

These basic services are implemented in IPsec by the use of the Encapsulating Security Payload (ESP) [46] and the Authentication Header (AH) [45]. ESP provides confidentiality by encrypting the original IP packet, building an ESP header, and putting the ciphertext in the ESP payload.

The AH can be used alone for authentication and integrity-checking if confidentiality is not an issue. With AH, the static fields of the IP header and the data have a hash algorithm applied to compute a keyed digest. The receiver uses its key to compute and compare the digest to make sure the packet is unaltered and the sender's identity is authenticated.

2.5.2 IPSec Modes

Both AH an ESP support two modes of use: transport and tunnel mode.

Transport mode provides protection primarily for upper-layer protocols. The protection extends to the payload of an IP packet. Typically, transport mode is used for end-to-end communication between two hosts. As shown in Figure 2.8, ESP in transport mode encrypts and optionally authenticates the IP payload but not the IP header. As shown 2.9, AH in transport mode authenticates the IP payload and selected portions of the IP header.

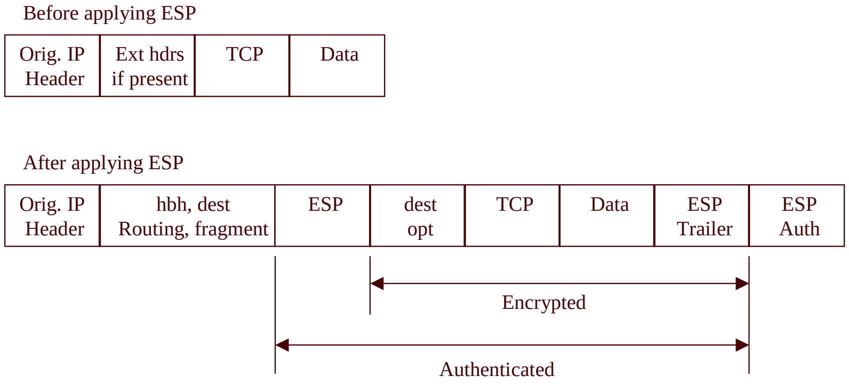


Figure 2.8: ESP Transport Mode of IPv6 [46]

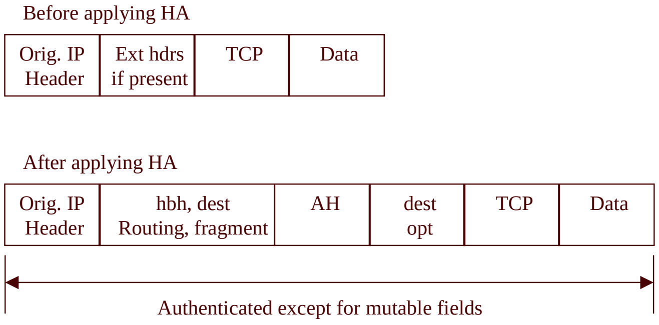


Figure 2.9: AH Transport Mode of IPv6 [45]

As shown in Figures 2.10 and 2.11, Tunnel mode provides protection to the entire IP packet. After the AH or ESP fields are added to the IP packet, the entire packet plus security fields is treated as the payload of new "outer" IP packet, with a new outer IP header. The entire original, or inner, packet travels through a "tunnel" from one point of an IP network to another; no routers along the way are able to examine the inner IP header. Tunnel mode is used when one or both ends of an SA is a security gateway. In this thesis, IPSec tunnel mode is applied to protect signaling and data traffic over the wireless channel.

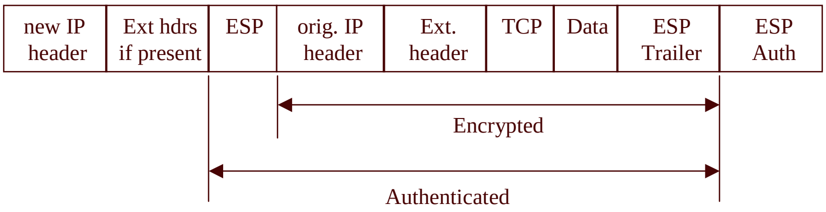


Figure 2.10: ESP Tunnel Mode of IPv6 [46]

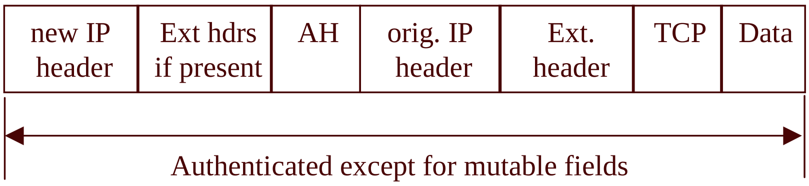


Figure 2.11: AH Tunnel Mode of IPv6 [45]

2.5.3 Security Associations

The building block on which secure communications are built is a concept known as a security association. Security associations relate a specific set of security parameters to a type of traffic. With data protected by IP Security, a separate security association exists at least for each direction and for each header type, AH or ESP. The information contained in the security association includes the IP addresses of the communicating parties, a unique identifier known as the Security Parameters Index (SPI), the algorithms selected for authentication or encryption, the authentication and encryption keys, and the key lifetimes as shown in Figure 2.12.

An SA normally includes the following parameters:

- *[required]:*
  - Authentication algorithm and algorithm mode being used with AH

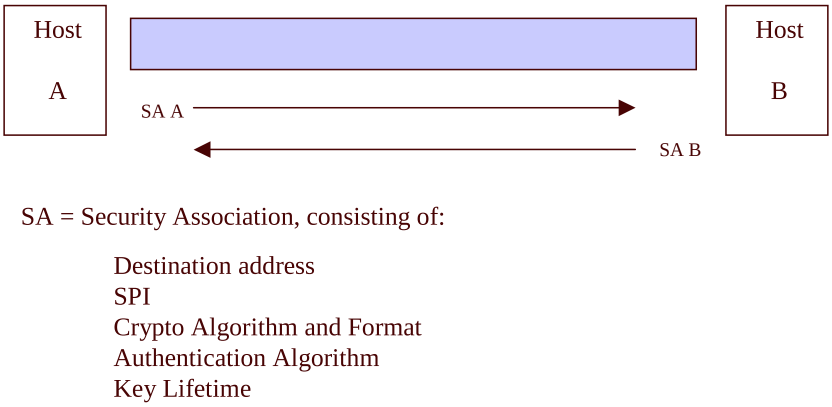


Figure 2.12: The Establishment of a Secure Tunnel between Hosts A and B

  - Key(s) used with the authentication algorithm in use with AH
  - Encryption algorithm, algorithm mode, and transform being used with ESP if encryption is used
  - Key(s) used with encryption algorithm in use with ESP if encryption is used
  - Presence/absence and size of a cryptographic synchronization or initialization vector field for the encryption algorithm.

- *[optional]:*

  - Lifetime of the key or time when key change should occur.
  - Source address(es) of the SA, might be a wild-card address if more than one sending system shares the same SA with the destination.
  - Sensitivity level (for example: Secret or Unclassified) of the protected data [required for all system claiming to provide multi-level security, recommended for all other systems].

A SA is unidirectional. An authenticated communication session between two hosts will normally have two SPIs in use (one in each direction).

The encryption and authentication algorithms are directly responsible for the strength the security the system can provide. There are however major drawbacks in this area. As the Internet is an global network, the IP should provide uniform security everywhere. Many countries, however, either restrict or forbid the use, or export of encryption algorithms. This means that the IPSec must be able to balance between the legal restrictions in use of strong encryption and authentication, and the one that is available everywhere.

All hosts claiming to provide IPSec services must implement the AH with at least the MD5 algorithm using a 128-bit key as specified in the AH RFC. An implementation may support other authentication algorithms in addition to keyed MD5. All ESP implementations must support the use of the Data Encryption Standard (DES) in Cipher-Block Chaining (CBC) mode as detailed in the ESP specification. Other cryptographic algorithms and modes may also be implemented in addition to this mandatory algorithm and mode. MD5 and DES-CBC should be set as default algorithms.

2.5.4 Key Management

The key management portion of IPSec involves the determination and distribution of secret keys. A typical requirement is four keys for communication between two applications: transmit and receive pairs for both AH and ESP.

Manual and automated are two types of key management. Oakley [71] is a refinement of the Diffie-Hellman key exchange algorithm; ISAKMP [59] does not dictate a specific key exchange algorithm. It defines procedures and packet formats to establish, negotiate, modify, and delete security associations.

The Internet Key Exchange (IKE) [34] describes a hybrid protocol using part of Oakley and part of SKEME [53] in conjunction with ISAKMP to obtain authenticated keying material for use with ISAKMP, and for other security associations such as AH and ESP for the IETF IPsec DOI. The purpose is to negotiate, and provide authenticated keying material for security associations in a protected manner.

2.6 Authorization

2.6.1 An Overview of Authorization

Authorization is defined as the act of determining if a particular right, such as access to some resource, can be granted to the presenter of a particular credential.

Generally, authorization can be categorized in two major classes [79]:

- **Authentication-based** mechanisms require an authentication of the subject as precondition for the authorization. The information for the authorization decision is stored as a policy, such as in Access Control Lists (ACLs) of operating systems in the form "User U is allowed to perform action A on an object O".

  Authorization policies define those actions a subject is permitted to perform on an object. An authorization policy may be positive (permitting) or negative (prohibiting).

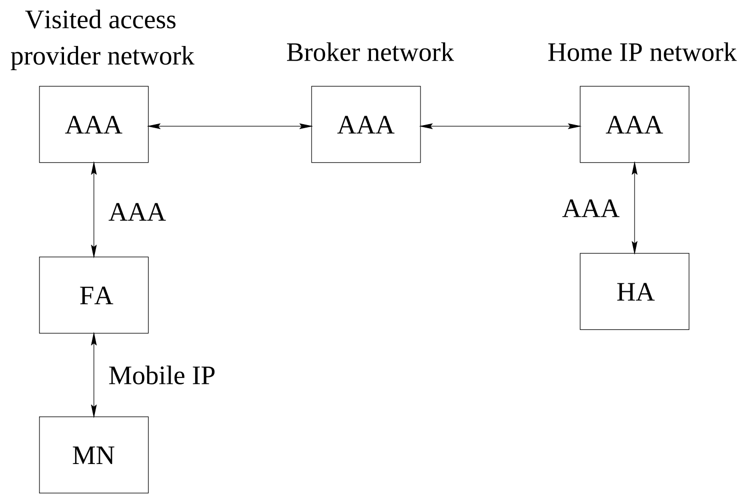


Figure 2.13: Mobile IP and AAA [98]

- **Credential-based** mechanisms use credentials which are trustworthy information being hold by subjects of an authorization process. Credential-based mechanisms are widely accepted in e-business, e.g., in form of micro-payments Millicent or used in SPKI [70].

  The credential-based mechanism has a great similarity with the authentication-based mechanism. The credential has a similar form as a policy, whereby the set of objects has only one element which is the user (may be anonymized) who owns the credential.

2.6.2 Authorization in IP Mobility Environment

In the IP mobility environment, authorization is achieved by the interaction between the Mobile IP and an AAA protocol.

Figure 2.13 provides an example of a MIP network that includes an AAA infrastructure.

Following is the registration actions after MN appears within the a foreign network:

- MN issues a registration to FA.

- FA sends an AAA request to AAAL including authentication information and the registration request.

- AAAL determines whether the request can be satisfied locally through the use of the Network Access Identifier (NAI) which has the format of user@realm. AAAL can use the realm portion of the NAI to identify the Mobile Node's home AAA Server. If AAAL does not share any security association with the MN's AAAH, it may forward the request to its broker.

- The broker which shares an SA with AAAH will forward the request.
- AAAH receives the AAA request and authenticates the MN since it has a security relationship with it. Afterwards, it begins to authorize the request.
- The authorization includes the generation of dynamic session keys to be distributed among all mobility agents, optional dynamic assignment of a home agent, optional dynamic assignment of a home address (HA) (this could be done by home agent also) and optional assignment of QoS parameters for the MN [37].
- Once authorization is complete, the AAAH issues an unsolicited AAA request to the home agent including the information in the original AAA request as well as the authorization information generated by the AAAH.
- The home agent generates a registration reply that is sent back to the AAAH in an AAA response. The message is forwarded to AAAL via the broker and finally arrives at the FA which provides services to MN.

Concerning minimized latency involved in getting wireless (cellular) access to the network, only one single traversal is needed to authenticate the user, perform authorization and process the registration request. AAAL maintains session state information based on the authorization information. If the MN moves to another FA within the foreign domain, a request to the AAAL can immediately be done in order to immediately return the keys that were issued to the previous FA. This minimizes an additional round trip through the Internet when micro mobility is involved, and enables smooth handover.

The key distribution issue is one of our interests. After the MN is authenticated, the stage of authorizing the AAA requests includes the generation of three session keys which to be shared between MN and HA, MN and FA and FA and HA respectively. Each key is propagated to its related mobility entity through either the AAA protocol or MIP. Once the session keys have been established, the mobility entities can communicate without the AAA infrastructure. However the session keys have limited lifetimes, they need to be updated after expiration.

2.7 DoS Attacks

2.7.1 An Overview of DoS

In general, Denial of Service (DoS) is defined as to prevent or inhibit the normal use or management of communications facilities.

A DoS attack can occur in different forms:

- **destruction or alteration of configuration information:** an attacker may modify e.g. routing packets so that the complete routing tables of a single or multiple nodes are corrupted;
- **consumption of scarce resources:** an attacker may send a large number of bogus requests to a network. A network consumes so much of its resources such as network connectivity, bandwidth and CPU cycles etc. that its performance is significantly degraded;
- **invalidation of scarce resources:** an attacker may reserve resources of a network so as to make the resources unavailable to legitimate users. This may result from a failed access control.

The last two items are related to the focus of the thesis.
Generally, the common measurements against DoS attacks include

- **"Backscatter" Measurements of Flooding Attacks:** Most attacking tools choose the source addresses at random for each packet sent. As a consequence, the responses are distributed also randomly and almost equally among the Internet. While monitoring a sufficient number of hosts in the Internet, it is possible to detect and evaluate the backscatter;
- **Filtering rules:** enabling filtering rules against IP proofing with a firewall;
- **Indication of DoS attacks:** using administrative tools to monitor the load state. For example, if there are two many connections in the state "SYN_RECEIVED" may indicate that the system is being attacked;
- **Authentication and Access Control:** before consuming resources on a request or granting access to a requester, a network checks the authenticity of the request, and tailors its access right to the request.

2.7.2 DoS in IP Mobility Environment

Two kinds of DoS attacks can be identified in IP mobility environment: resource exhaustion DoS and DoS due to access control failure.

The resource exhaustion DoS was addressed in the design of MIPv6 [43]. Corresponding Node (CN) does not maintain any state about Mobile Node (MN) until an authentic Binding Update arrives. This is a way to protect network nodes against memory exhaustion DoS attacks. Also only symmetric cryptography is used to protect Corresponding Node (CN) to be relatively safe against CPU resource exhaustion DoS attacks [43]. In addition to the memory exhaustion and CPU resource exhaustion DoS attacks, a large number of bogus registration request can cause also the signaling capacity of the links in an access network to be degraded considerably.

DoS attacks can occur if access control fails. If an adversary is granted access to the resources of an access network mistakingly, it can send malicious requests to reserve the resources. Thus an access network may have insufficient resources for legitimate requests.

2.8 Summary

In this chapter, the related background information in the context of secure and efficient QoS-aware mobility support in IP based cellular networks was given.

Regarding mobility, the evolution from IPv4 to IPv6 were described. The problems when IP was used in mobility environment were identified and the introduction of Mobile IP was motivated. The mobility support in IPv4 and IPv6 networks were described. Mobility scenarios were classified into the global mobility and local mobility. Since mobility issues are not among the focuses of the thesis, HMIPv6 was selected as the micro mobility management protocol in the thesis.

Regarding QoS, two main approaches - IntServ and DiffServ were discussed. To meet the requirement of QoS-aware mobility, the two approaches are considered in the protocol design in the thesis.

To provide the background information for the security services of authentication, confidentiality and data integrity, cryptographic algorithms were introduced. In a mobility environment, the joint AAA and Mobile IP architecture were described for the purpose of authentication and authorization.

Since IPSec was proposed to protect data over the wireless channel in the network layer, it was described in terms of usage modes, security association and key management.

Since DoS attack is one of the security concerns in the thesis, the general forms of the DoS attack and the principle countermeasures to the threats were described.

Chapter 3

Problem Statement

In this chapter, the problems in a QoS-aware mobility support for real-time applications will be stated. At first, the general operations during a (re-)registration procedure will be introduced. Then the importance to minimize the latency for the latency-sensitive applications in a (re-)registration procedure will be discussed. Afterwards, the threat of the DoS attacks in the QoS-aware mobility scenarios is discussed. Finally, the requirements to support secure and efficient QoS-aware mobility are identified.

3.1 General Operations In a (Re-)registration Procedure

Generally, a handover procedure consists of two phases: IP connectivity establishment and (re-)registration [1]

In the phase of IP connectivity establishment, a Mobile Node (MN) configures a stateless link-local IP address and performs a Duplicate Address Detection (DAD) to ensure the uniqueness of the address. This phase may take around $80ms$ according to [52].

The focus of this thesis lies in the second phase - after the IP connectivity establishment. In such a phase, in addition to the binding update operation, the network should provide QoS support to applications. The functionality which reserves desirable **forwarding treatment** to a particular packet stream requires to establish **session states** along a end-to-end communication path. There are two kinds of session states:

- the state at an Access Router (AR) is used for packet classification, packet metering and packet marking parameters in order to provide Differentiated Services (DiffServ) [12];

[1] A registration procedure happens in inter-domain handovers whereas a re-registration procedure happens in intra-domain handovers.

- the state along the whole path (which includes the AR) indicates e.g. the committed bandwidth to ensure the forwarding treatment in order to support Integrated Services (IntServ) [101].

Above all, before taking resource/time consuming actions on a registration request, or committing resources to an applications, the network checks the authenticity of the request. Authentication is the prerequisite of any other operations including other security checks. After the authentication check, the network also needs to verify whether the requested QoS is what the user is entitled to use (i.e. authorization).

After a successful binding update operation, a new security association should be established to protect the data packet especially between the MN and the new AR.

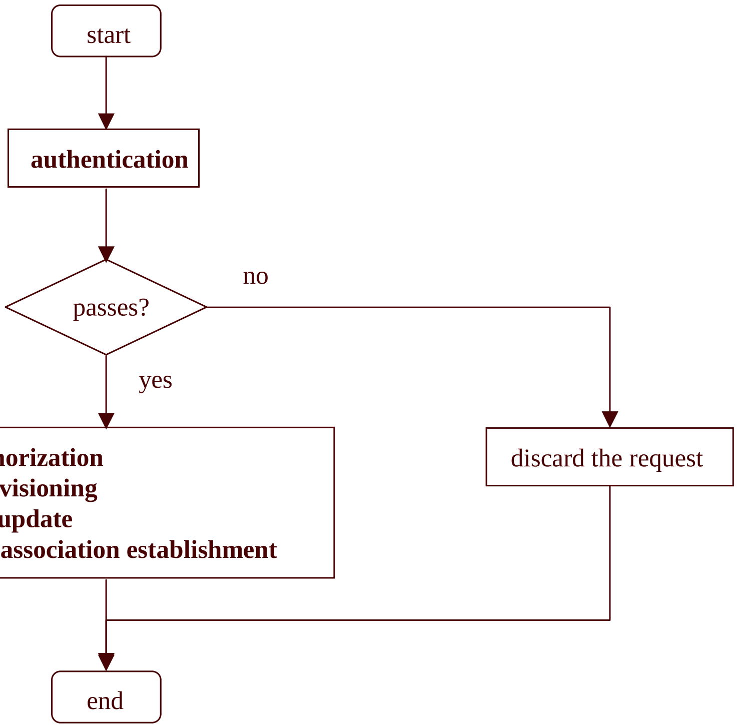


Figure 3.1: The General Operations in the (Re-)registration of a Handover Procedure

In summary, as shown in Figure 3.1, in the (re-)registration phase of a handover procedure, authentication must be done first to protect the access network against DoS attacks. If the authentication check passes, the operations which may demand significant network's resources are then performed.

Table 3.1: Specifications of Some Typical Real-time Applications [54, 97]

Application	Type of Data	Target Bit Rate	Max Delay
One-way Voice	audio	a few - tens of Kbps	150 ms
Video Telephony	static video	$\leq 500\ Kbps$	200 ms)
Digital TV	high motion video	$\leq 1\ Mbps$	$\leq 1\ sec$
Video Conference	natural text	512 $Kbps$	$\leq 1\ sec$

3.2 (Re-)registration Procedure with Latency-sensitive Applications

For the latency-sensitive applications such as Voice over IP (VoIP), **seamless connectivity** is crucial [52]. Therefore, in a (re-)registration procedure, all the operations including security checks, QoS provisioning, mobility management and Security Association (SA) establishment should introduce minimal latency.

Real-time applications, which demand the typical operations of packetization of a stream, transmission of the packet flow and de-packetization at a distant sink, are loss-tolerant and delay-sensitive. The applications do not work properly if packets are delayed for more than a tolerated amount of time.

Generally, real-time applications with **two-way communication** require a low delay of $100s$ of ms [11, 54] in order to ensure the interactivity of the applications. The real-time applications with **one-way communication** require a minimal delay. Within the minimal delay, any packets buffered in the play-out buffer are used to reconstruct the original stream [20, 78]. Packets arriving after the minimal delay become useless.

Table 3.1 lists the requirements of some typical real-time applications. A minimum of bandwidth at each node on the communication path should be provided in order to meet the requirement of the target bit rate.

Figure 3.2 shows an example of one-way communication (i.e. a receiver is receiving H.263 video streams) during a handover procedure with the registration operations. During the procedure, the operations as shown in Figure 3.1 must be done. In the interruption period, some frames maybe re-sent and some maybe buffered. Therefore, in such a mobility scenario, a (re-)registration procedure should introduce a minimal delay so that the perceived quality of the applications is not impaired.

Regarding the operations in a (re-)registration procedure, the following two issues should also be considered in addition to the requirement of **minimal latency**:

- In a registration procedure, the authorization data of a mobile node is transferred to the access network so that the access network is able to perform re-authorization without involving the mobile node's home domain. Thus, the latency can be minimized in a re-registration procedure. However, it should to avoid unnecessary

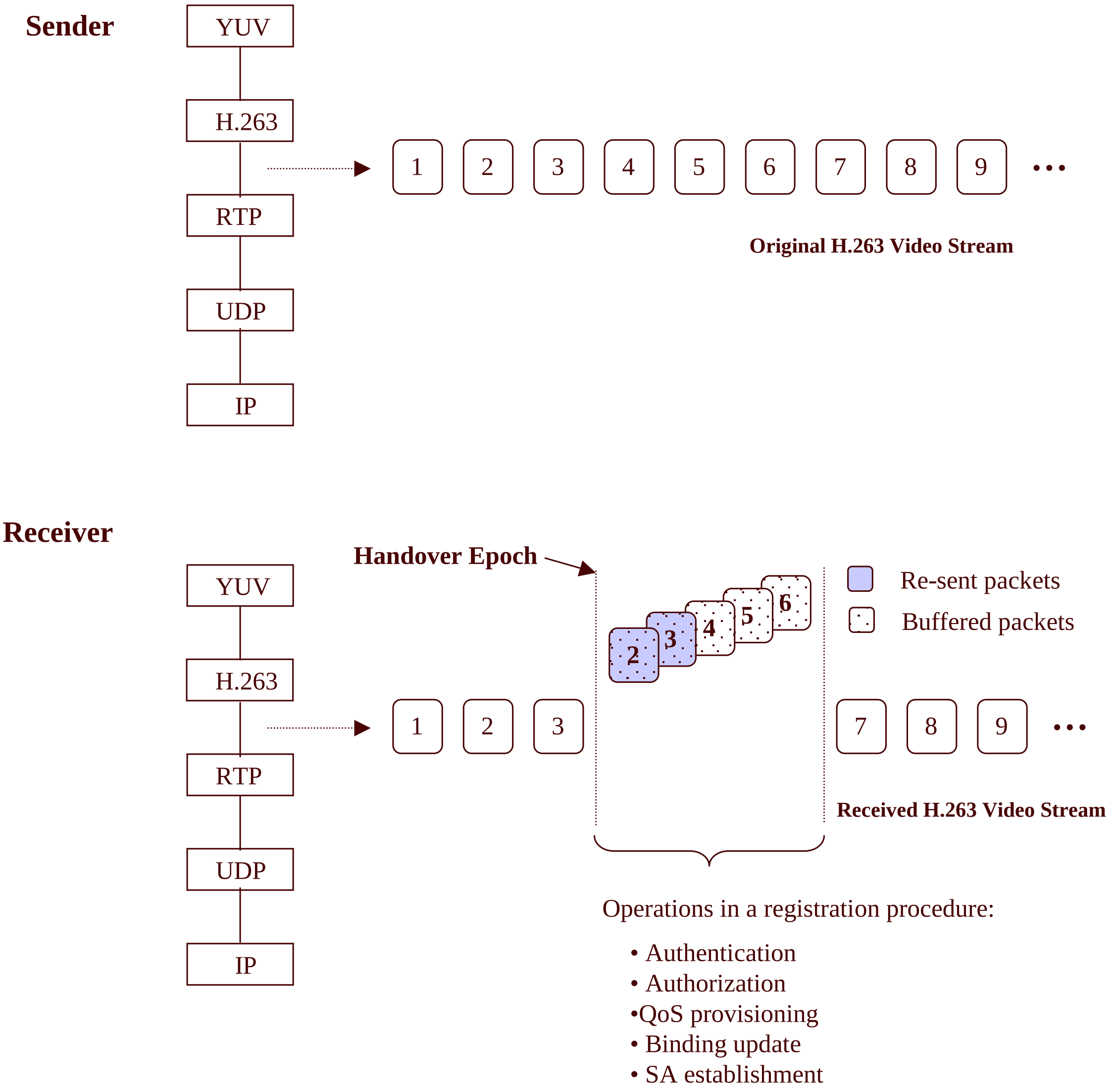


Figure 3.2: An Example of Registration Procedure with H.263 Stream

disclosure of a mobile user's subscribed values of QoS parameters as the authorization data in the access network since these values may be regarded as confidential information;

- protection of the signaling for the SA establishment and the user data should use different keys in order to reduce vulnerability caused by a compromised key.

3.3 Denial of Service in QoS-aware IP Mobility

QoS mechanisms which aim to guarantee certain service characteristics in networks supporting mobile communications give rise to DoS threats: these mechanisms could be abused by malicious entities launching so-called Denial of Service (DoS) attacks, which aim to reduce the availability of services to legitimate users. If the network can not efficiently check the "credibility" of a QoS-request during a handover process, malicious entities could flood the network with bogus QoS-requests in order to cause the exhaustion of the available resources by temporal reservations. This represents one specific DoS threat.

A simple solution could be realized with the following procedure: when an access router (AR) receives a QoS request, before starting the resource reservation process, the AR communicates with a local security authority, e.g. a local AAA server (AAAL), to authenticate the MN and authorize the QoS request. Only when the check passes, the path reserves resources according to the request. Obviously, the latency introduced by proceeding to security checks at the AAAL, which includes the contribution of the propagation delay and processing time at AAAL, is not desirable when low latency of the registration process is a major concern. Moreover, the same checks at an AAA server have to be performed on all the bogus requests from attackers. Thus all the security check signaling may degrade the performance of the access network substantially by depleting the signaling capacity of the path between the AR and the AAAL and exhausting the computing resource of the AAAL. This represents another specific DoS threat [18].

3.4 Requirements

Identifying the requirements for a secure efficient QoS-aware mobility support is essential in protocol design. Based on the discussion above, the requirements are summarized as follows:

- **Various operations:**

  A (re-)registration procedure should include the following operations: authentication, authorization, QoS provisioning, binding update and SA establishment. QoS provisioning include QoS path set up and DiffServ support.

  - **QoS path set up:** in order to guarantee bandwidth at each involved node for a particular session, session state should be set up along the new QoS path during a handover.
  - **DiffServ support:** Connection state for appropriate treatment to a specific session should be re-established at an edge router in the access network during handovers.

- **Minimal latency:**

A (re-)registration procedure should introduce minimal latency during handovers. The latency results from both an individual operation and the integration of the operations in a complete procedure.

- **Avoidance of disclosure of the subscribed QoS Values:**

  When MN's home AAA server gets involved in authorization, it should not disclose unnecessarily the subscribed QoS values which are regarded as the MN's confidential information.

- **Separation of keys in the SA establishment:**

  A SA should be established efficiently to protect the data between MN and the new AR. The corresponding signaling should be protected with a key different from the key to protect the user data.

  Moreover, the data between MN and the old AR should be protected with a different SA from that for data protection between MN and the new AR. If both SAs use the same key, when the key for the SA between MN and the old AR were compromised, vulnerability would be introduce to the SA between MN and new AR.

- **Protection against DoS:**

  Before the access network commits its resources (e.g. computing power, storage space and signaling capacity) on a request, authentication should be performed first.

  Authentication may introduce vulnerability to DoS attacks itself. Authentication should be light-weighted in terms of cryptographic computation.

3.5 Summary

In this chapter, the requirements to support a secure and efficient QoS-aware mobility was introduced.

The question that this thesis attempts to answer is

how can a QoS-aware mobility in IP based networks be achieved in an secure and efficient manner?

This question is important to answer based on the following considerations:

- **Involved Operations:** in a QoS-aware (re-)registration procedure, in addition to the operation for mobility management, some actions also need to be taken such as (re-)authentication, (re-)authorization, QoS provisioning and SA establishment.

- **Low-latency in (re-)registration procedure:** To keep active communications uninterrupted during a handover, the registration latency should be minimized.

- **Robustness against DoS attacks:** Consuming minimum resources such as signaling capacity and computing power for QoS signaling is favorable not only for networks, but also for mobile hosts to guarantee high throughput. This is true especially when the network is facing a great amount of malicious QoS requests - DoS attacks.

- **Security Considerations:**

  An access network should always maintain a balance between the two conflicting requirements: efficiency and security. The cost of providing efficient QoS-aware mobility support should be evaluated in terms of security.

  In addition to the security goals identified in the MIPv6 design [43], some security issues were identified for a secure and efficient QoS-aware mobility support such as avoidance of disclosure of confidential data in an access network and separation of keys for signaling and data protections.

In the following chapter, the design space will be discussed to meet the above mentioned requirements. Also the related work will also be described.

Chapter 4

Design Space and Related Work

To meet the identified requirements in Section 3.4, this chapter first discusses the design space for a secure and efficient QoS-aware support in IP mobility. Then related work in the design space will be introduced. Finally, the design oneline for the formulated proposal will be specified based on the discussion of the design space and related work.

4.1 Design Space

The discussion on “design space” is organized as the following:

1. first, how to provide the **security** services in the QoS-aware mobility environment, such as authentication, authorization, data protection over wireless, and network protection against DoS;

2. secondly, how to provide the **QoS** services such as IntServ and DiffServ in the IP mobility environment;

3. finally, how to perform the operations related to the two kinds of services (i.e. security and QoS) jointly in an **efficient** manner.

4.1.1 Design Space of Security Schemes

The security design includes authentication, authorization, data protection over wireless channel, and DoS protection.

Authentication

When MN presents a request to an AR (termed as *new AR* in the access network, the network should check the authenticity of the request first.

The *network-layer* authentication can be performed by either by the **AR** or a **local security authority**.

- **Authentication Check By AR:**

  In case that **access routers** perform the authentication check, when receiving an authenticator from the MN, the *new AR* verifies the authenticator based on

  - the relevant security information like a *key* transmitted from the AR which the MN previously associates with (termed as *old AR*). (This is adapted in Context Transfer (CT) which will be introduced in Section 4.2.13); or
  - the security information transmitted from a local security authority (e.g. AAAL); or
  - its own knowledge of some security information. For example, when the authenticator is encrypted with a *key* which is known to all ARs in the access network, the *new AR* can verify the authenticator without obtaining any information from other entities.

- **Authentication Check By A Local Security Authority:**

  In case that a **local security authority** performs the authentication check, when receiving an authenticator from the MN, the *new AR* forwards it to the local authority i.e. Local AAA Server (AAAL). The idea was adapted in two related work: *PANA* and *Diameter MIPv6 Application*, which will be introduced in Section 4.2.1 and Section 4.2.2 respectively.

Authorization

The authorization check is always integrated with the authentication check. Similar to authentication, authorization can be performed by either the *new AR* or *a local security authority*.

- **Authorization Check By AR:**

  The authorization information can be presented to the *new AR* by either an entity in the access network or the MN itself. Two related works use the idea: *Context Transfer (CT)* and *token-based* approach, which will be introduced in Section 4.2.13 and Section 4.2.3 respectively.

- **Authorization Check By A Local Security Authority:**

  The authorization check can be done by using an AAA protocol such as “Diameter”. The operations shown in Figure 4.3 can be applied for authorization. When receiving a request from MN, AR sends a Diameter message to e.g. AAAL for the authentication and authorization checks. The authorization data is conveyed as the *Authorization Attribute Value Pair (AVP)* in the Diameter messages.

Additionally, an AAA server may get involved in a *Service Level Agreement (SLA)* based authorization procedure. A related work using *SLA* will be described in Section 4.2.4.

Data Protection Over Wireless Channel

Since the wireless channel is the most vulnerable part in an end-to-end communication, some measures need to be taken to protect both signaling and data traffic over the wireless channel.

Similar to authentication, the data protection over wireless channel can be done at different layers. For example, *Wired Equivalent Privacy (WEP)* was proposed to offer authentication, confidentiality and data integrity services in the IEEE 802.11 environment at the link layer. However, some security problems have been identified when *WEP* is used to provide the security services [99, 5]. The improvement on IEEE 802.11 security by integrating IEEE 802.1X - IEEE 802.11i has been approved recently in IEEE 802.11 Task Group I.

At the network layer or above, the data protection over wireless channel can be provided by *IPSec* or *TLS*.

- *IPSec*: IPSec can be used to protect user data over wireless channel. [74] documents the proposal to establish an *IPsec* Security Association (SA) and the *IPsec* based access control.

  The Internet Key Exchange (IKE) scheme [34] is used for establishing the IPsec SA between the MN (termed as PaC) and the AR (termed as EP). It is proposed that MN and AR use the “session ID” and “Key ID” to identify the pre-shared key.

  $$ID\_KEY\_ID_{data} = (Session - Id|Key - Id)$$

  As a result, an ESP tunnel mode is used for protecting the traffic between MN and AR in a IPSec SA.

- *TLS*: TLS was adapted in the design of CASP QoS Client Protocol which will be introduced in Section 4.2.10.

DoS Protection

If *authentication* is performed by means of verification of a digital signature, the verifier may consume a relatively large amount of resources on it. Moreover, if AR sends a *authentication* request to the AAAL server whenever it receives a user’s request, extensive bogus requests may degrade the signaling capacity in an access network significantly. Therefore, *authentication* may introduce the risk of Denial of Service (DoS).

In order to prevent the DoS threats in the above mentioned scenarios, a two-step authentication is necessitated. A *weak authentication* is used as the first step authentication, followed by a *stronger authentication* as the second step [60]. The *weak authentication* should involve a cryptographic algorithm which requires easy computation. The algorithm may be the symmetric algorithm or the HMAC algorithm.

Two approaches can be used as the first step authentication: *client puzzle* or *cookie*. Client puzzle will be introduced in Section 4.2.5 and cookie usage in Mobile IP will be introduced in Section 4.2.6.

4.1.2 Design Space of QoS Schemes

The QoS design is classified into RSVP-similar approach for IntServ Support, policy-based approach for DiffServ Support and QoS brokers for IntServ and DiffServ support.

RSVP-Similar Approach for IntServ Support

Resource ReSerVation Protocol (RSVP) [15] is a network control protocol that allows Internet applications to obtain special Quality of Service (QoS) for their data flows. RSVP was designed without the issue to support scenarios with high mobility. The addresses of the hosts were assumed to be fixed and route changes were considered as an exception.

In RSVP, every reservation needs therefore a two-way message exchange. A simultaneous reservation setup in uplink and downlink direction with only one message exchange is not possible.

RSVP was originally designed to support multicast applications. An RSVP router can merge different data flows based on the information in the *Path* message. It was not designed for unicast applications.

RSVP is often considered as being to complex and therefore error-prone. Therefore, some issues raise concern when RSVP is used in the mobility scenarios. For example, when an MN changes locations, the handover may force a change of its assigned IP addresses. RSVP uses the destination address of a message flow as identifier. As a consequence the filters associated with a reservation are not able to identify the flow anymore and the resource reservation is ineffective, until a refresh with a new set of filters is initialized. [4].

There have been several designs for extensions to RSVP to address the above mentioned issues. *Mobile RSVP* will be introduced in Section 4.2.7, *MIPv6+RSVP* will be introduced in Section 4.2.8, and the *QoS-conditionalized BU* approach will be introduced in Section 4.2.9, and *CASP QoS Client Protocol* will be introduced in Section 4.2.10.

Policy-based Approach for DiffServ Support

The general solution for DiffServ support is to enforce policies at the edge router (i.e. AR) to filter data traffic from and to the MN to pass through.

One example has been shown in Figure 4.5, in which the policies are transmitted to the access router in the form of *SLA*. Then the router translates the policies into router rules, allowing particular *treatment* to the user packet.

QoS Brokers for IntServ and DiffServ Support

QoS Broker (QoSB) is an network entity which can be used for network resource management. It can perform resource allocation and resource release for IntServ support, and admission control for DiffServ support. The idea was adapted in *Moby Dick* project which will be introduced in Section 4.2.11.

4.1.3 Design Space of Solutions to the Efficiency Issue

Some ideas were proposed to perform the operations of security, QoS and mobility management efficiently:

- **Fast Handovers:** allowing an MN to form a new IP address before it attaches to the new AR, and buffering and forwarding packets from old AR to new AR before packets can arrive at the new IP address directly. The idea will be introduced in Section 4.2.12.

- **Context Transfers:** transferring session states regarding e.g. QoS and security data from old AR to new AR. It will be introduced in Section 4.2.13.

- **Tight Coupling and Processing in Parallel:** integrating different operation processes if possible; performing different operations in parallel if possible.

  In order to improve the behavior of reservation-based QoS, as defined in the IntServ, the QoS signaling and mobility signaling can be coupled in HMIPv6 network during an intra-domain handover. Three levels of coupling are discussed as follows [58]:

  - *De-coupled*

    In this mode, QoS and micro-mobility mechanisms operate independently of each other and the QoS implementation is not dependent on a particular mobility mechanism. QoS of the session will not be resumed until a new reservation is complete in the path from the across-over router and the new access router. The old path between the cross over router and the old access router cannot be explicitly removed until the lifetime times out.

- *Loosely Coupled*

  The loosely coupled approach uses mobility events to trigger the generation of QoS signaling once valid routing information has been installed in the network and can be localized to the path from Cross-over Router (CR) to the nAR.

- *Tightly Coupled*

  The closely coupled uses the same signaling mechanism to propagate the mobility and QoS information. This approach minimizes the disruption to traffic streams after handover by ensuring that the reservation is set up as soon as possible after installing routing and QoS information simultaneously in a local area.

Coupling the QoS and micro-mobility mechanisms provides definite performance advantages over the normal de-coupled approach. In the HMIPv6 network, the *tightly coupled* (also denoted as *QoS+BU*) mode can provide improvements in performance in terms of efficiency during a intra-domain handover.

Besides the condition that the new path can satisfy the requested QoS, successful authorization is another condition for a successful QoS-aware handover. Therefore, a re-authorization process must be performed in a handover procedure. In order to achieve an efficient handover, the *QoS+BU* process and the *re-authorization* process can undergo in parallel.

4.2 Related Work

All the related work mentioned in the discussion of the design space will be described in this section.

4.2.1 PANA

In the Protocol for Carrying Authentication for Network Access (PANA) approach [28], an *Authentication Server (AS)* acts as the local authority.

PANA is designed as a generic framework to provide a link-layer independent authentication service for access control. PANA carries Extensible Authentication Protocol (EAP) [2] which can adapt various authentication algorithms and protocols. The framework PANA is shown in Figure 4.1.

After configuring a new IPv6 address, a MN (termed as PaC) first sends a *PAA Discovery* message to find a PANA Authentication Agent (PAA) in the access network since it is assumed that the MN is new to the *new AR* (termed as Enforcement Point (EP)). After discovering PAA, it starts *PAA Authentication* by the Extensible Authentication

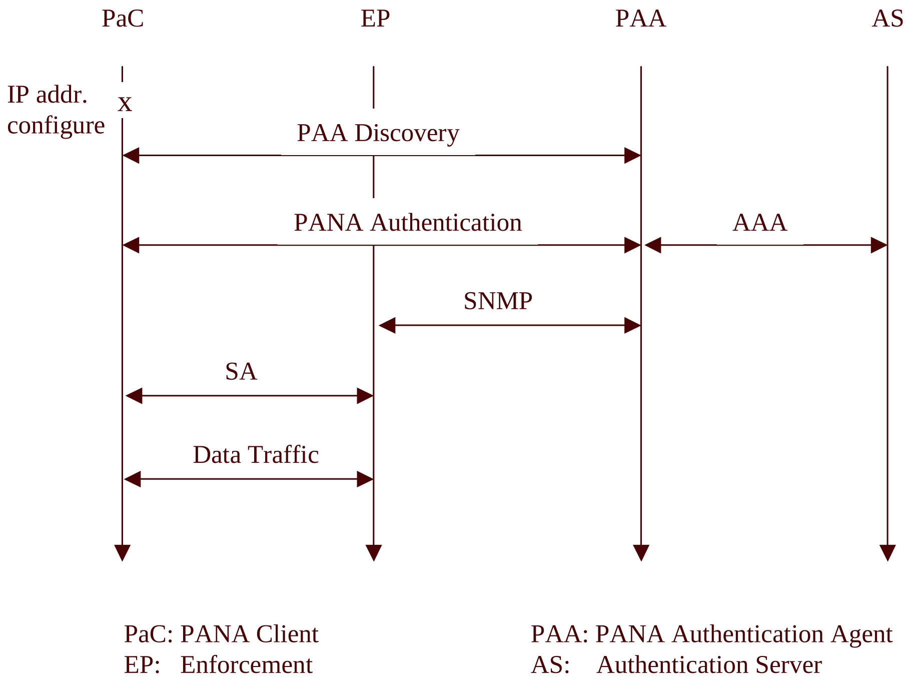


Figure 4.1: PANA Framework [56]

Protocol (EAP) exchange with the PAA. The PAA initiates an authentication process with the Authentication Server (AS) using an AAA protocol.

Upon receiving the authentication and authorization result from the AS, the PAA informs the MN about the result of its request. Meanwhile, the PAA sends the MN's specific attributes such as cryptographic keys and access control policy to the *new AR* by using Simple Network Management Protocol (SNMP).

A Security Association (SA) has been established between the MN and *new AR*. MN's data traffic can be protected under the SA.

The PANA approach can provide authentication and authorization by using an AAA protocol or Context Transfer Protocol. However, it does not address the problem of disclosing the confidential authorization data.

The first phase of the PANA protocol run is discovery and initial handshake, in which a MN discovers the next authentication agent and negotiates with it. This phase may introduce a great latency which is not favorable to a registration procedure which demands low latency.

When using EAP, authentication signaling is protected by the Security Association (SA) between MN and the authentication server. This key is different from that used by IPSec, which can be enabled in PANA to protect user data over wireless channel. Therefore, the signaling and data are protected by different keys. However, in handling mobility, PANA does not address to use different keys for the old SA (i.e. SA between MN and the old AR) and the new SA (i.e. SA between MN and the new SA).

PANA uses the cookie mechanism to prevent the DoS attack of memory allocation with a single message [28].

The issues PANA does not address include QoS provisioning for IntServ support and mobility management.

4.2.2 Diameter MIPv6 Application

In the approach of Diameter Application in MIPv6 [25], a local AAA server serves as the security local authority. It enables to authenticate and authorize a mobile node in order to give it access to network resources by carrying the AAA information and the binding update information in one message.

Figure 4.2 shows involved entities and Figure 4.3 illustrates a typical registration procedure.

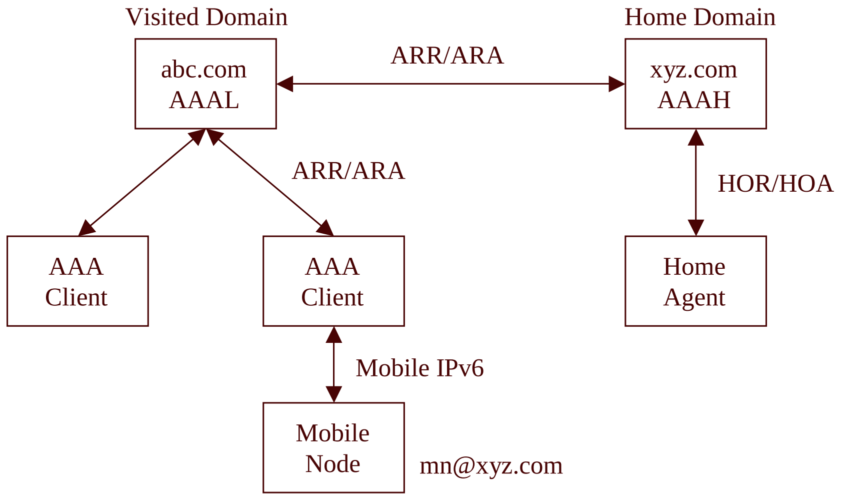


Figure 4.2: Involved Entities in Diameter Mobile IPv6 Application [25]

1. When entering a new network or at power up, the MN listens to the router advertisements and retrieves the local challenge, the visited network identifier, and the information to derive a CoA.

2. MN generates a new CoA based on the network's information. It composes a registration request message destined to the AR which acts as an AAA client in the AAA infrastructure.

3. The AAA Client first verifies the freshness of the request thanks to the local challenge contained in it. If successful, it performs Duplicate Address Detection (DAD) and creates a Diameter ARR (AA-Registration-Request) message. It then sends the message to AAAL.

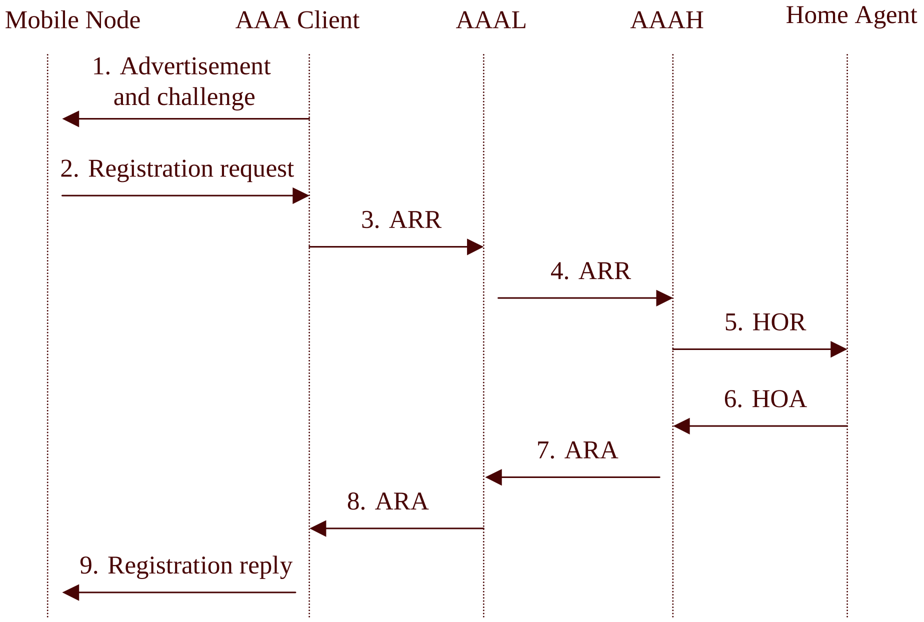


Figure 4.3: Message Flow of Registration Procedure of Diameter MIPv6 Application [25]

4. When AAAL receives an ARR message, first it verifies if the message is coming from a valid AAA Client. Then it checks the MIPv6 Feature Vector Attribute Value Pair (AVP). Finally, it sends the ARR to the MN's home AAA server.

5. When receiving an ARR message from an AAAL, AAAH first verifies the message is coming from a valid AAAL. Security associations between AAA server are assumed to be pre-existing. AAAH then authenticates the user using the NAI provided by the MN as MN identity. If the mobile node is successfully authenticated, AAAH also computes some network authentication data based on the host challenge and eventually other information depending on the authentication algorithm adopted. AAAH sends a HOR (Home-Agent-MIPv6-Request) message to the HA including a newly created binding update. It also sends some security keying material to allow the home agent to compute the key(s) for the security association between the MN and the home agent to authenticate future binding updates.

6. Home agent creates a binding cache and computes the key(s) for the security association with the MN from the data received. It also generates a binding acknowledgement message to be sent encapsulated to the MN. HA sends a HOA (Home-Agent-MIPv6-Answer) message to AAAH including the Binding Ack.

7. AAAH may also compute other keying material according to the keys requested by the MN and send it to the MN passing through the AAAL. AAAH then send an

ARA (AA-Registration-Answer) message to AAAL including the MIP-binding-acknowledgement Attribute Value Pair (AVP) if the MN sent an embedded Binding Update (BU) or request for a Home Agent (HA).

8. When receiving a ARA message from AAAH, the AAAL may optionally store locally information contained in the AVPs of the message received from AAAH (e.g. authorization information, session keys, etc.) and then forwards the message to the AAA client.

9. When receiving an ARA message from AAAL, the AAA client converts the message to the appropriate protocol to the MN; this message carries the authentication data, binding acknowledgement, keying material to set up the different session keys. It sends the corresponding registration reply message to the MN.

The Diameter MIPv6 Application approach can be used for authentication and authorization in mobility scenarios. It tries to minimize latency in a registration procedure by embedding binding update information in the AAA signaling. However, it does not introduce the micro mobility schemes to minimize handover latency during intra-domain handovers.

During an intra-domain handover, in addition to be fully responsible for authentication and authorization with involving MN's home AAA server [24], the local AAA server is also responsible for deploying Security Association (SA) between MN and the new AR. In such a case, even though signaling and data are protected with different keys, the key for data protection over wireless channel between MN and the old AR is same as that for data protection between MN and the new AR. Thus, if the key between MN and the old AR is compromised, vulnerability is introduced to the SA between MN and the new AR.

Authenticating requests at the local AAA server via the AAA protocol may introduce threats of DoS attacks.

Moreover, it does not address QoS provisioning and protection of confidential authorization data to be disclosed in access network.

4.2.3 A Token-based Approach

A *token based* approach can be used MN to provide its own authorization information. That is, when the token is authenticated, the authorization data is the token is also authenticated. Thus the authorization data can be used to authorize the request.

The token approach is illustrated as follows.

A token in the context is used as an optimization mechanism that provides fast re-authentication and re-authorization when the mobile node roams within the same administrative domain. An authorization token is a Cryptographic Message Syntax (CMS)

[40] encapsulated (digitally signed or encrypted) collection of objects. An authorization token consists of the following fields [32, 33]:

- AUTH_ENT_ID: The unique identifier of the entity which authorized the session
- SESSION_ID: Unique identifier for this session
- SOURCE_ADDR: Address specification for the session originator
- DEST_ADDR: Address specification for the session destination
- START_TIME: The starting time for the session
- END_TIME: The end time for the session
- RESOURCES: The resources which the user is authorized to request
- AUTHENTICATION_DATA: Authentication data of the session authorization policy element

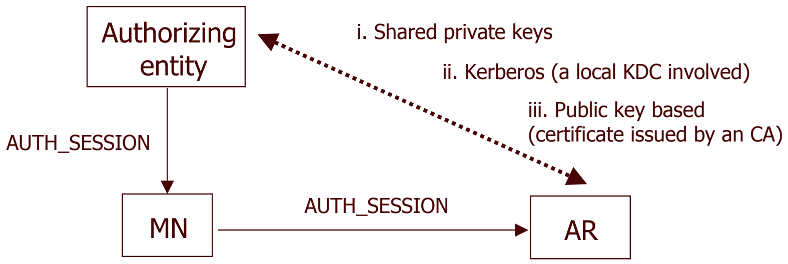

Authorizing entity
i. Shared private keys
ii. Kerberos (a local KDC involved)
iii. Public key based (certificate issued by an CA)
AUTH_SESSION
MN
AUTH_SESSION
AR

Figure 4.4: An Example Operation of Authorization Token

Figure 4.4 illustrates an example operation of authorization token, which enables per-session admission control via a session authorization policy element [32, 33].

An authorizing entity first issues a MN a session authorization policy element token, denoted as *AUTH_SESSION*, authorizing an access to the resources for a session. Then the MN inserts the *AUTH_SESSION* element in the resource reservation message to an AR. AR thus verifies the authenticity of the request and processes the resource reservation message based on admission policy. The authenticity can be done by three approaches:

- **Shared symmetric keys:** This assumes that the *Authorizing Entity* and the *AR* shares a symmetric key and with policies detailing which cryptographic Al algorithm to be used for the authentication.

- **Kerberos:** It is assumed that a Kerberos Distribution Center (KDC) is present to support the Kerberos authentication. When receiving a resource reservation request, *AR* contacts the local KDC to request credentials for the *authorizing entity*. The *KDC* responds with the credentials consisting of a *ticket* for the *authorizing entity* and a temporary encryption key (also termed as a *session key*). The *ticket* is used to access the authorizing entity and the *session key*, which is now shared by the *AR* and the *authorizing entity*, is used to authenticate the two entities each other. Thus, the two entities communicate to authenticate the request from MN.

- **Public key:** In a public key environment, it is assumed that the *authorizing entity* has a key pair - *private key* and *public key*. The *private key* is secured at the *authorizing entity*, whereas *public key* is stored in digital certificates and a trusted party - Certificate Authority (CA) which issues these digital certificates.

  The *authorizing entity* uses its private key to generate *AUTHENTICATION_DATA*. AR uses the authorizing entity's public key, which is stored in a *digital certificate* to verify the *AUTHENTICATION_DATA*. The certificate can be *X.509 V3 digital certificates* or *Pretty Good Privacy (PGP) digital certificates*.

Since the *AUTHENTICATION_DATA* field contains the authentication data of the *AUTH_SESSION* policy element and signs all the data in the policy element up to the *AUTHENTICATION_DATA*, the verification of the authenticity of the request passes, the integrity of the policy element can be ensured.

In summary, the token based approach addresses the issues of authentication and authorization. It is inferred from the design of CASP Client Protocol [85] that a token can be used for QoS path set up and policy deployment to support DiffServ.

When AR receives the *AUTH_SESSION* token sent by MN, it first performs authentication. That performing authentication before any other operations may be useful against DoS attacks. But the Kerberos and public key based approaches for authentication may introduce DoS themselves since many message exchanges are required.

It does not address mobility management, avoidance of disclosure of confidential authorization data and use of separated keys.

4.2.4 Service Level Agreement

SLA of a mobile user is a set of agreements between a mobile user and a Internet Service Provider (ISP). To translate *SLA* into device-dependent configuration is done through policies [27]. A policy may include one or more conditions and actions. Thus, when the conditions are evaluated to be true, the AR takes actions to enable the access control on MN's traffic.

Figure 4.5 shows an example of a SLA based access management approach.

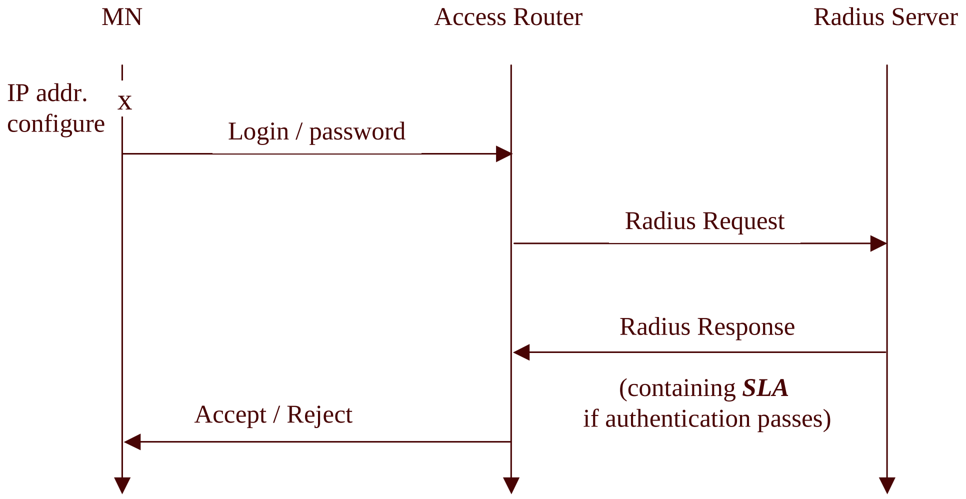


Figure 4.5: An Example of a SLA Based Access Management Approach [27]

When a MN arrives in an access network, it obtains an IPv4/IPv6 address using e.g. stateless configuration mechanism.

The MN needs to give its login name, password and its ISP in the registration request. Once the information is validated, the router sends a request to an authority (e.g. an Remote Access Dialing User Service (RADIUS) AAA server) for authentication and authorization checks. If the AA check is successful, the server responds with the user's *SLA* including the authorized traffic, associated bandwidth and authorized time. Then the router translates the corresponding policies into router rules, allowing the MN to access its requested services.

This approach supports authentication, authorization, Security Association (SA) establishment, and DiffServ. However, it has no mechanism to set up a QoS path. This approach has some potential problems:

- *The DoS Attack Threat:*

  Since the AA check has to be performed at the AAA server on all requests, attackers are able to generate extensive bogus requests to trigger so many such processes as to degrade the performances of the access network and the AAA server substantially by depleting the signaling capacity of the path between the AR and the AAAL and exhausting the computing resource of the AAAL. This represents a specific DoS threat.

- *SLA Confidential Data Exposure:*

  A premier user (whose permission is e.g. 10 Mbps) may not want to disclose its identity by exposing the content of his SLA (i.e. 10 Mbps) in a foreign access network , especially when he requests only a low QoS (e.g. 10 Kbps) to check his mail.

> Therefore, it is not desirable to disclose unnecessarily the overall service contract which includes service provisions the MN subscribes in its home network in the visited network since the contract content is somehow regarded as confidential information for an MN.

This approach does not address the mobility management. Also it does not address the requirement of low latency in a registration procedure.

4.2.5 Client Puzzle

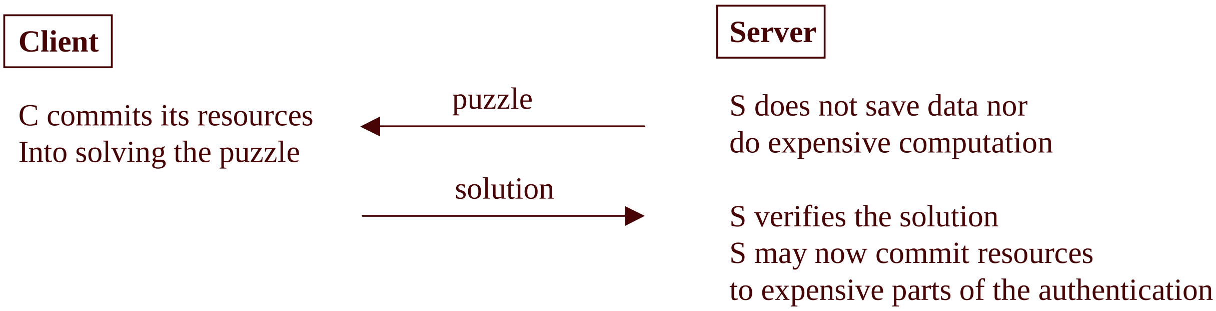


Figure 4.6: An Example Operations of the Client Puzzle Idea

The basic operations of the *client puzzle* approach is illustrated in Figure 4.6. When a client sends a request to a server, the server does not save any data nor do expensive computation. Instead, it sends a cryptographic puzzle to the client. The client has to commit its resources to solve the puzzle. Only after the server verifies that the solution from the client is correct, it may now commit its resources keeping state or doing expensive computation [6].

In one of the proposals of the client puzzle idea [6], the difficulty of the puzzle is controlled by the number of first digits which must be zero in a hash digest:

$$h(C, N_s, N_c, X) = 000...000Y$$

The more zeros there are, the more difficult for the client to find the solution. The denotations refer to Table 4.1.

Thus, the puzzle can prevent intensive connection initiations from attackers hence enhance the DoS-resistance of a server.

However, solving cryptographic puzzles also imposes a computational burden to any legitimate client and the server as well as requiring additional message exchanges. It would add non-trivial latency to setting up a connection between client and server. Therefore, it is not desirable in our application scenario where fast handover support is a major concern.

Table 4.1: Denotations in the Example Operations of the Client Puzzle Idea [6]

h	a cryptographic hash function (e.g. MD5 or SHA)
C	the client identity
N_s	the server's nonce
N_c	the client's nonce
X	the solution of the puzzle
k	the puzzle difficulty level
000...000	the k first bits of the hash value must be zero
Y	the rest of the hash value may be anything

4.2.6 Cookie Usage in Mobile IP

The concept of "cookie" in the context of client-server transactions started from "Netscape cookie", which were pieces of information generated by a Web server and stored in the user's computer ready for future access. The concept was developed to be a popular defense against Denial of Service (DoS) attacks due to the fact that verifying a cookie introduced low computational burden to a server. For example, TCP/IP in Linux uses cookies to defend against the *SYN* attack [9]. In [44], a *cookie* which was a light-weight authenticator was proposed to be presented in each message, and no further processing was needed on the message unless the verification of the cookie passed.

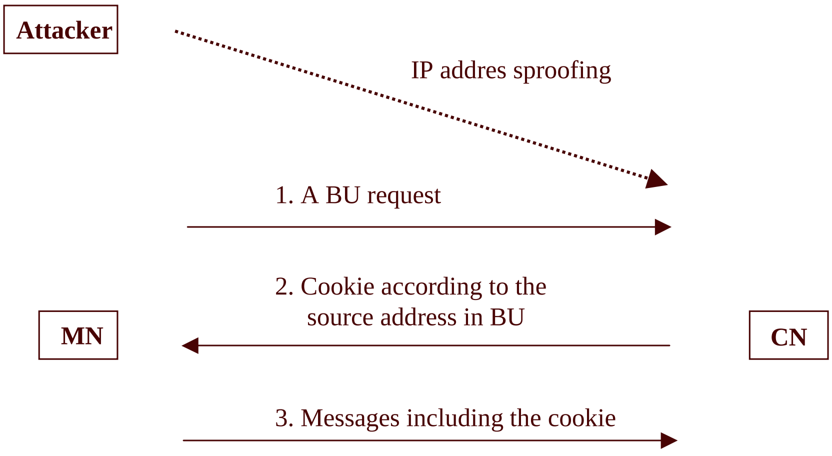

Attacker
IP addres sproofing
1. A BU request
2. Cookie according to the source address in BU
MN
CN
3. Messages including the cookie

Figure 4.7: The Cookie Concept in Mobile IP

The concept was adapted in Mobile IP design. As shown in Figure 4.7, when a MN sends a Binding Update (BU) request to a Corresponding Node (CN), the CN does not keep any state of this binding. It sends a cookie to the source address in the BU. If the preceding BU request is sent by the owner of the source address, i.e. the MN, it can continue the communication with CN by including the cookie in messages. Only the CN is able to verify the cookie so as to prevent DoS from attackers. If an attacker sends a BU request spoofing the MN's address, it can not receive the cookie hence it can not continue the communication.

In the above examples, cookies can be verified only by their generator. In a mobility scenario, for example MN moves from the *old AR* to the *new AR*, an entity (e.g. *new AR*), which is not the cookie generator, should be able to verify a cookie immediately when receiving a request, without preceding communications or existing information about the cookie. Therefore, if the cookie idea is used in defending against DoS, some modifications on the existing solutions are necessitated.

4.2.7 Mobile RSVP

Mobile RSVP (MRSVP) [95] is a protocol which extends RSVP in order to support mobility. It is based on the mechanism to reserve resources for a mobile node in advance before it performs a handover.

For making advance reservations, it is necessary to specify the set of access routers a mobile node will eventually visit. Each access router has the function of a proxy which reserves the resources for the mobile node with the proper parameters. The currently unused reservations are called passive reservations.

When the mobile node arrives at a particular access router, the path becomes active and the path to the previous one passive, so the data can still be delivered effectively. The active and passive reservations can be merged by calculating an effective flow specification. Thus, multiple reservations over long distances through the Internet can be avoided.

Since the reservations have been established before an MN determines an access router to perform a handover, this approach supports seamless QoS. However, it reserves resources unnecessarily at the access routers and paths which will not be used by the MN.

The protocol replies on the security mechanisms of RSVP. A mobile user must be authenticated before accessing resources of the network. The user authentication is useful to prevent DoS attacks. However, authentication itself may introduce DoS threat. The protocol does not address the issue. Also it does not address the issues of authorization and protection against DoS.

4.2.8 MIPv6+RSVP

A Mobile IPv6 and RSVP integration model was proposed in [19]. When the MN performs a handover, it gets a new CoA and subsequently sends a binding update to the CN. The CN then sends a PATH message associated with the new flow from CN to MN. Upon receiving this Path message, the MN replies with a RESV message immediately to reserve resources for the new flow. For each handover, the MN as a receiver has to wait for a new PATH message from the CN, it can only issue a new RESV message to the CN after getting the PATH message. However, all these RSVP re-negotiations are conducted end-to-end even though the path change may only affect a few intermediate routers. Hence, the long handover resource reservation delays and large signaling overheads caused by this end-to-end RSVP re-negotiation process could lead to notable service degradation in providing real-time services.

4.2.9 QoS-conditionalized BU

QoS-conditionalized Binding Update (QoSBU) [29] was designed to optimize the QoS-aware re-registration procedure of an intra-domain handover in HMIPv6-based networks. The main point is to piggyback QoS signaling on local binding update message being sent from MN to MAP.

A QoS hop-by-hop option piggybacked in the binding messages is used for QoS signaling. A handover takes place only upon the availability of sufficient resources along the new transmission path.

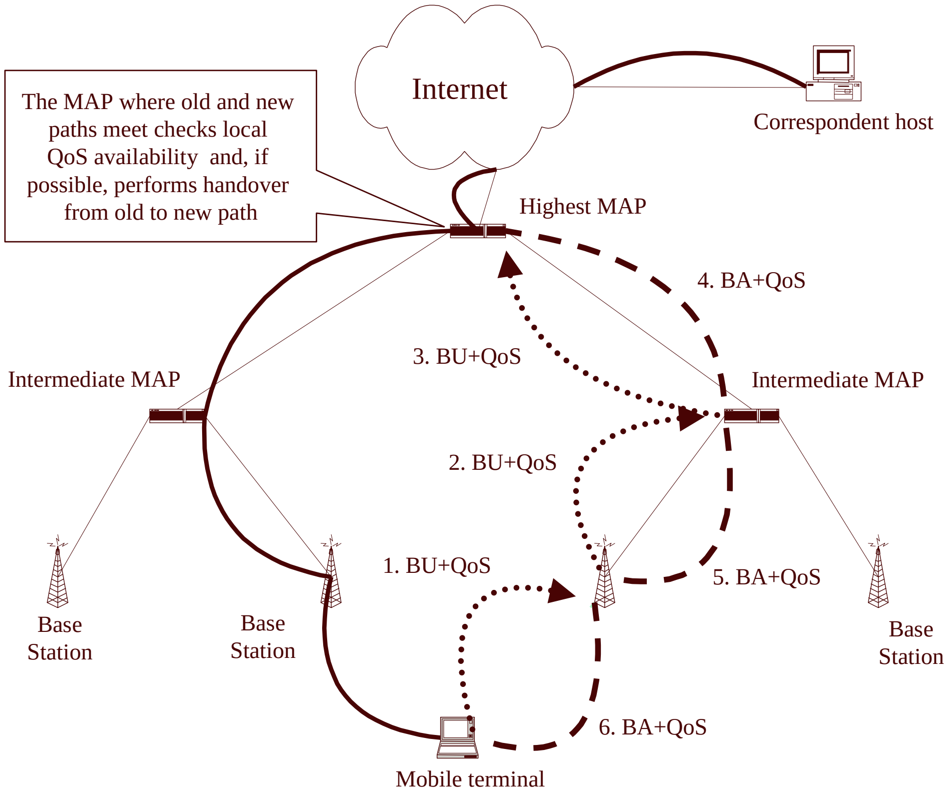


Figure 4.8: An Example Operation of the QoS-conditionalized BU Approach [29]

As shown in Figure 4.8, a QoS hop-by-hop option is carried in the message containing the BU option to the MAP – this message is called BU+QoS message. Each QoS entity between the MN and the MAP (including the MAP) will pass the QoS requirement represented by the QoS option to internal QoS mechanisms and check its resource availability. If resources are available locally, they are reserved and the message will be forwarded along its route. If resources are not available, negative feedback will be provided to the MN by means of an extended Binding Acknowledgement (BA+QoS) message. If a BU+QoS message has reached the switching MAP and passed the local QoS test as well, the binding update will take place (the binding cache in the MAP is updated to reflect the new LCoA) and a positive BA+QoS message is returned to the MN. Otherwise, no handover is performed and a negative BA+QoS message is returned to the MN. When observing a negative BA+QoS message, intermediate QoS entities can release reservations that could not be granted further upstream.

In summary, the QoSBU approach builds upon the hierarchical mobile IPv6 protocol and is especially fit for local mobility, where the signaling overhead is reduced. It also enables the mobile node to flexibly choose among a set of available access points so that the mobile node can transmit packets through a route which offers satisfying QoS for IntServ support. However, it does not address how to support DiffServ.

The approach uses an AAA infrastructure for authentication and authorization. But it does not consider the DoS threat.

Due to the fact that the QoS information is embedded in the IPv6 hop-by-hop extension header, this approach is not scalable to be used for an end-to-end QoS path establishment, it can not be used as a generic QoS signaling protocol. These issues were addressed by the two-layer signaling design in an IETF working group Next Steps in Signaling (NSIS). The CASP QoS Client Protocol is one typical design in the family.

4.2.10 CASP QoS Client Protocol

The Cross-Application Signaling Protocol (CASP) provides a generic signaling service by establishing state along the data path from a sender to one receiver for unicast data, or multiple receivers for multicast data [87]. As shown in Figure 4.9, the modular CASP framework includes a general purpose messaging layer (M-layer) and a number of client layers for various signaling applications.

A messaging layer (i.e. M-layer) session state consists of a session identifier, a flow identifier, a previous and next CASP hop, a refresh interval and a branch identifier. The session identifier is independent from the IP address of the sender and is therefore suited to identify the message flow from a mobile node even after a change of the care-of address due to a handover. The support by the CASP hops for different client layers is optional. An intermediate node is not required to support the client layers. The M-layer, referring to NSIS Transport Layer Protocol (NTLP) in the NSIS working group, is combined with a standard transport protocol like TCP or UDP. The different CASP clients

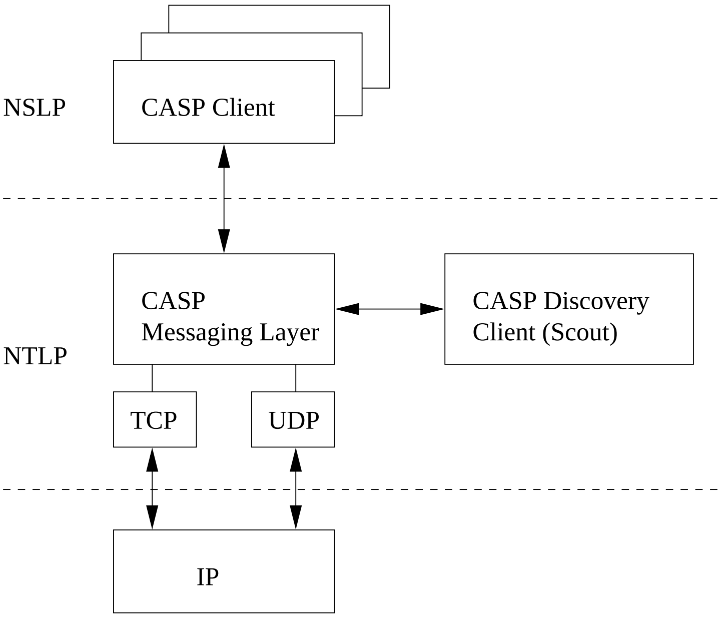


Figure 4.9: The CASP framework [87]

are allocated to the NSIS Signaling Layer Protocol (NSLP) in the two-layer design in NSIS.

In CASP design, signaling and discovery message delivery are separated. The *Scout protocol* is used to discover the next suitable CASP node and the required soft-state refresh interval if the next CASP node is more than one network-layer hop away. It is only needed in case that no other suitable means of discovering the next CASP node are available. In environments with high mobility, however, the discovery process with scout will increase a considerable overall handover latency.

CASP QoS client [85] is a client protocol for CASP. It offers per-flow resource allocation and reservation. A process for resource allocation and reservation may contain one of the five message types: *Query, Reserve, Commit, Release and Success*.

The mobility issue is being discussed in the NSIS working group [89, 30]. In a mobility environment, the CASP QoS Client Protocol was designed to be able to react on route changes in a more comfortable way. It only requires the creation of a reservation along the new path until the merge point (i.e. the crossover router) with the old path is reached. The scope of a reservation can be restricted to the new part of the path by generating and deleting the appropriate signaling messages at the affected nodes. The separation of the Session ID and the Traffic Selector enables the merge point to associate an existing reservation with the ID provided by the incoming signaling message.

In the design of CASP QoS Client Protocol, *TLS* can be used to set a SA between MN and AR.

Figure 4.10 shows a general message exchange of the CASP QoS Client protocol. From the figure, it is observed that a TLS setup may take 4-6 message exchanges. Ob-

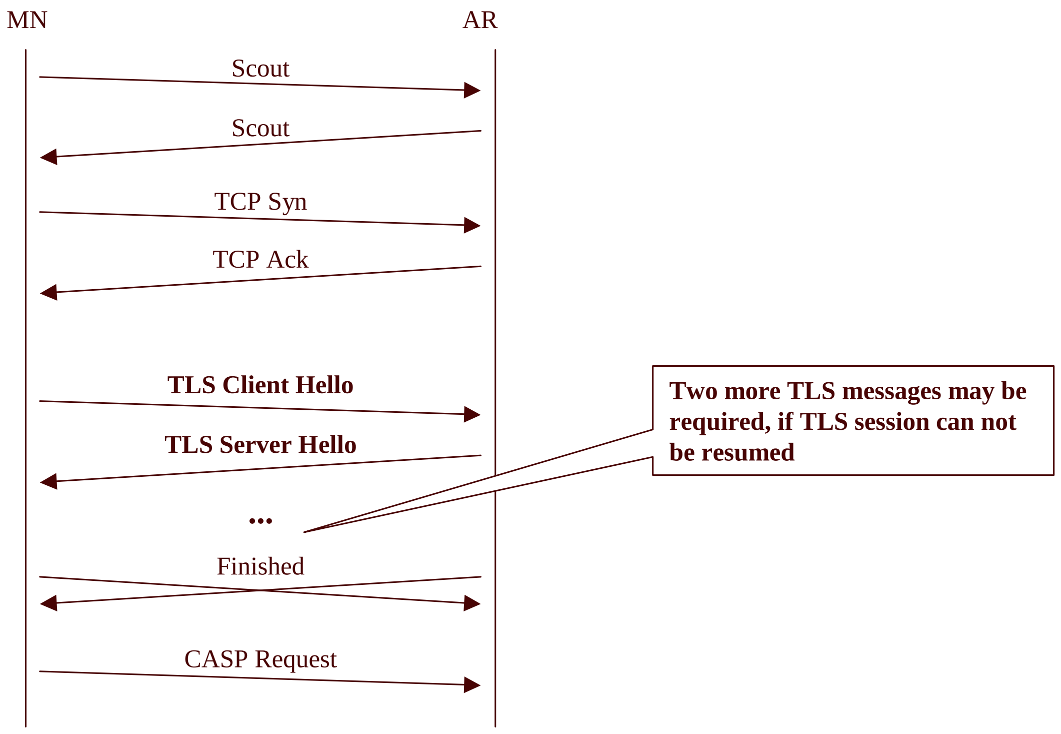


Figure 4.10: The General Message Exchange of CASP QoS Client Protocol

viously, the process is not able to provide seamless handover support.

Moreover, there is no proposal to address the seamless handover procedure featuring Authentication and Authorization (AA) checks, IntServ and DiffServ support and protection against DoS. As one of the drawback in the CASP QoS Client protocol design, the surplus message exchanges were shown in Figure 4.10.

4.2.11 Moby Dick Project

In the Moby Dick project [62], a QoS Broker (QoSB) is used to take admission control decisions and configure all access network components according to a set of conditions given by the administrative entity. The QoS Broker has the information of all nodes (including access routers and intermediate routers) and links in the access network. Figure 4.11 shows the functionalities of a QoS broker, which include allocating resources all the nodes involved in IntServ support for a session; enabling admission control in an access router; and interacting with AAAL for Authentication, Authorization, Accounting (AAA) purposes.

In addition to QoS provisioning, Moby Dick can cooperate with Diameter MIPv6. Therefore, it provides authentication, authorization, and partial optimization in a registration procedure as discussed on the Diameter MIPv6 protocol. However, MN's home AAA server distributes MN's SLA to the visited network, disclosing the MN's confi-

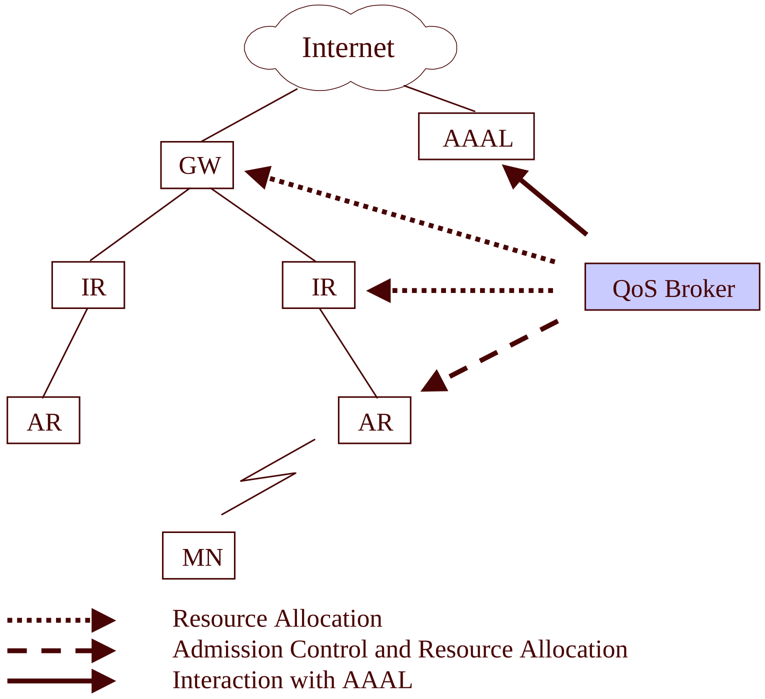


Figure 4.11: Functionalities of a QoS Broker

dential data unnecessarily. Also the AAA signaling for authentication may introduce DoS attacks.

4.2.12 Fast Handovers

Fast Handovers are required to ensure that the layer 3 (Mobile IP) handover delay is minimized, thus also minimizing and possibly eliminating the period of service disruption which normally occurs when a mobile node moves between two ARs. This period of service disruption usually occurs due to the time required by the mobile node to update its Home Agent (HA) using Binding Updates after it moves between ARs. During this time period the mobile node cannot resume or continue communications. This mechanism allows the anticipation of the layer 3 handover such that data traffic can be redirected to the mobile node's new location before it moves there [51].

Figure 4.12 illustrates a sequence of the message flows in the Fast Mobile IPv6 Handover protocol. The process consists of four phases [51, 23]:

- **Phase 1: Router Solicitation for Proxy and Proxy Solicitation Advertisement**

  Either the MN or the old AR (oAR) can initiate the Fast Handover procedure by using the L2 triggering information.

  - If the L2 triggering information is received at MN (Mobile initiated handover), MN sends a *Router Solicitation for Proxy (RtSolPr)* message to oAR

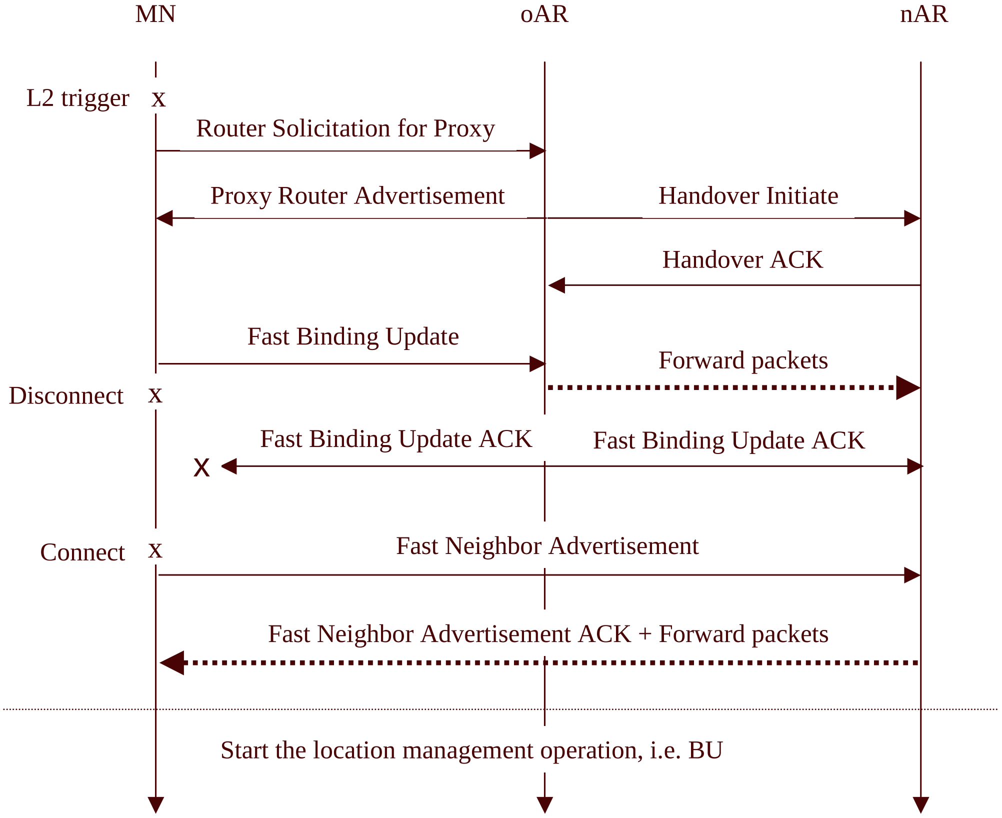

MN
oAR
nAR
L2 trigger
x
Router Solicitation for Proxy
Proxy Router Advertisement
Handover Initiate
Handover ACK
Fast Binding Update
Forward packets
Disconnect
x
Fast Binding Update ACK
Fast Binding Update ACK
X
Connect
x
Fast Neighbor Advertisement
Fast Neighbor Advertisement ACK + Forward packets
Start the location management operation, i.e. BU

Figure 4.12: Fast Mobile IPv6 Handover Protocol [51]

initiating an L3 handover. oAR will respond with a *Proxy Router Advertisement (PrRtAdv)* message. This message includes the network prefix information of the new AR (nAR) which will be used by MN to configure a new IPv6 address.

– If the L2 triggering information is received at oAR (Network initiated handover), oAR sends the *PrRtAdv* message to MN without the solicitation message.

Note that the *PrRtAdv* is essentially a router advertisement message from nAR which is the handover target. The advertisement carries the information for MN to form the new IPv6 address.

- **Phase 2: Handover Initiate and Handover Acknowledgement**

  The old AR (termed as "oAR") needs to negotiate the validity of the new IPv6 address with the new AR (termed as "nAR") on behalf of MN. oAR sends a *Handover Initiate (HI)* message, including either the new IPv6 address of the MN directly or the link-layer address of the MN which nAR can use to compute the new IPv6 address.

In the corresponding *Handover Acknowledgement (HACK)* message, nAR includes the result of negotiating the new IPv6 address. In case the new IPv6 address is not permitted, nAR creates a host-specific entry allowing the use of previous IPv6 address for the MN on its link until the MN obtains a correct IPv6 address.

After the negotiation, a forwarding path is set up for the packets destined to the MN's previous IPv6 address.

- **Phase 3: Fast Binding Update and Fast Binding Acknowledgement**

  To enable the forwarding path, MN needs to send a *Fast Binding Update (F-BU)* message to oAR. oAR is not allowed to use the forwarding path until it is authorized by MN. After receiving the *F-BU*, oAR starts forwarding packets to nAR and stops forwarding packets to the MN on its old link.

  At the time when nAR receives the forwarded packets, MN may have not yet established connectivity with it. In such a case, nAR needs to buffer the packets until it is able to forward the packets to MN's new IPv6 address.

  MN needs to receive *Fast Binding Update Acknowledgement (F-BACK)* from oAR to use the new IPv6 address. However, it is possible for MN to lose its connectivity with oAR before receiving the *F-BACK*. Hence, oAR should also send the message to nAR so that nAR can buffer the *F-BACK* or any packet meant for MN's previous IPv6 address for packet delivery once the MN establishes a new connectivity with it.

- **Phase 4: Fast Neighboring Advertisement and Fast Neighboring Advertisement**

  In case MN does not receive the *F-BACK*, it does not know whether its new IPv6 address is valid. To ensure the uniqueness of the new IPv6 address, MN sends nAR a *Fast Neighboring Advertisement (FNA)* which includes MN's old and new IPv6 addresses. nAR responds a *Fast Neighboring Advertisement acknowledgement (FNA-ACK)* and start forwarding the buffered packets and the packets being forwarding from oAR continuously to MN.

  In order to enable the packet being routed directly to MN's new IPv6 address, the location management operations (i.e. binding update) is then initiated.

In case MN can receive *router advertisement* message directly from nAR by means of e.g. channel scanning, MN is able to form a stateless IPv6 address without involving oAR. To check the uniqueness of the new IPv6 address and enable a forwarding path, MN can send the *F-BU* to oAR without the message exchange in Phase 1. Subsequently, oAR negotiates validity of MN's new IPv6 address and establishes the forwarding path with nAR by taking the operations in Phase 2.

The simplified sequence of the message flows is shown in Figure 4.13. In the performance analysis chapter, this model is used.

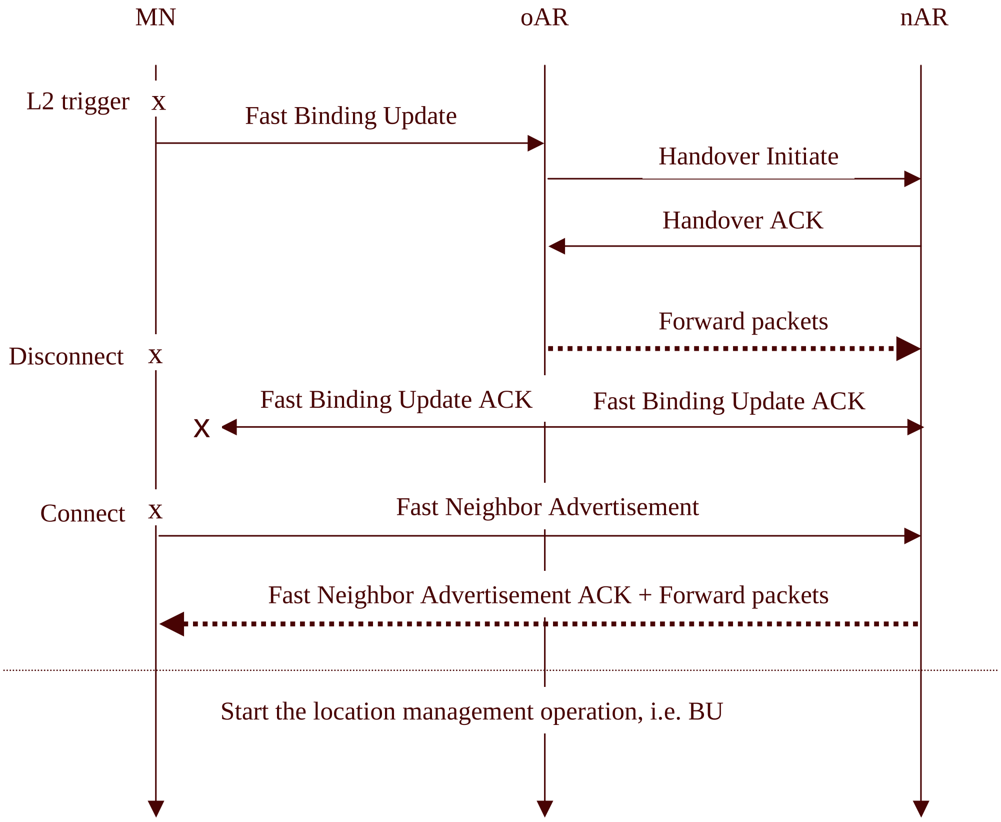


Figure 4.13: A Simplified Fast Mobile IPv6 Handover Protocol

An integration of *Fast Mobile IPv6 Handovers* and *HMIPv6* were proposed as shown in [91].

As shown in Figure 4.14, the HI_/HACK messages now occur between the MAP and nAR, checking the validity of the new IPv6 address and establishing a temporary forwarding tunnel. Therefore, the functionality at new AR in *Fast Mobile IPv6 Handovers* scheme is moved to Mobile Anchor Point (MAP) in the HMIPv6 infrastructure.

The models of MIPv6 and MIPv6+HMIPv6 will be used in the performance analysis in the corresponding chapter.

In summary, the essences of the Fast Handover Protocol are:

- **new IPv6 address:** MN obtains a new IPv6 address even when it is connecting with the old AR. Thus MN is able to start sending and receiving IP packets immediately once establishing a connectivity with the new AR.

- **forwarding path:** A forwarding path is set up between the old AR and the new AR. Hence no packet will be lost after MN loses its connectivity with the old AR.

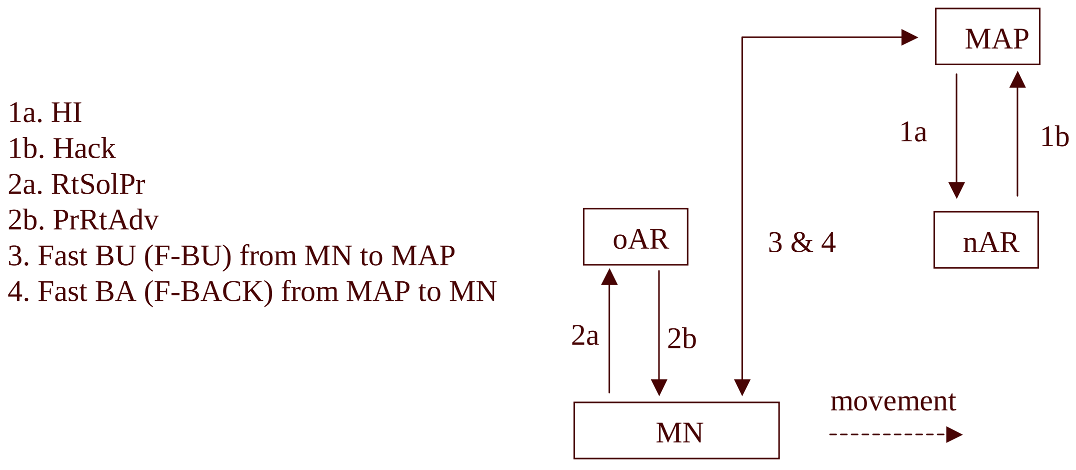


Figure 4.14: Fast Mobile IPv6 Handover Protocol using HMIPv6 [91]

However, the forwarding mechanism has some issues. For any real-time applications, if packets are buffered for a longer time than the admissible end-to-end delay, they can become useless [72]. Moreover, there is no QoS guarantee for the forwarded packets since the QoS signaling is not complete during the forwarding period. Also the packets are forwarded from nAR to MN without protection before a SA is set up.

Therefore, for the real-time applications which are delay-sensitive, minimizing registration latency is more important than reducing packet loss. Moreover, in a QoS-aware handover procedure, the validity of the new IPv6 address does not mean the success of the handover. The success of the handover also depends on whether the path can meet the requested QoS, whether the requested QoS is authorized to be used by the requester, whether the requester is authenticated.

4.2.13 Context Transfer

The service context can be transferred from the old AR to the new AR to enable a quick re-establishment of session states. The transfer of service context is designed to minimize the impact of host mobility [57].

The service context that could utilize a context transfer solution include Authentication, Authorization, Accounting (AAA), QoS and header compression information. Therefore, CT can be used to re-established session states efficiently in the intra-domain handover scenarios. The detailed protocol operations are documented in [57].

Context Transfer protocol was proposed to integrate with the *Fast Handover* protocol in [51, 52].

The sequence includes phases as shown in Figure 4.15.

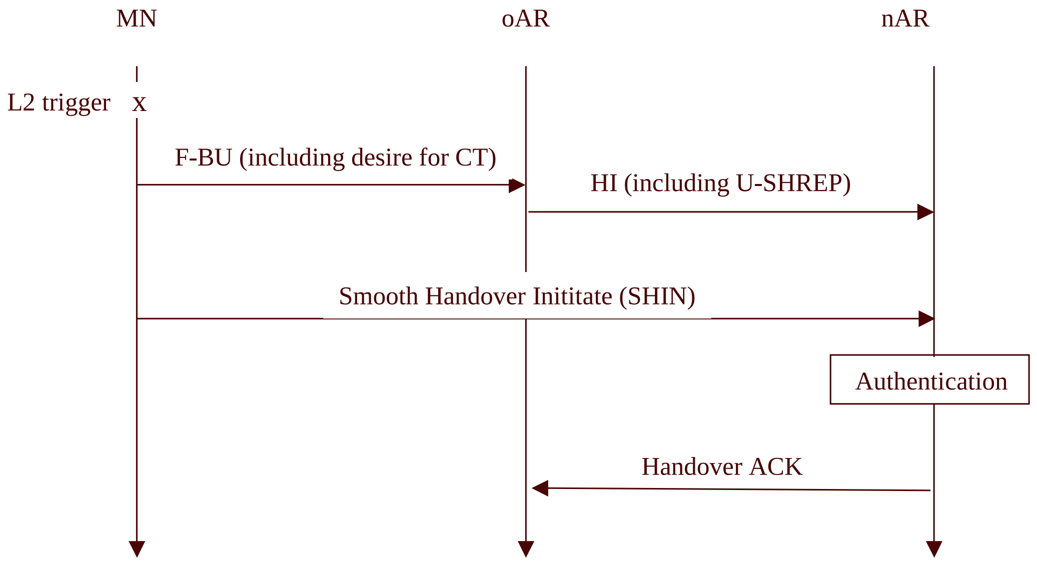


Figure 4.15: Context Transfers with Handover Signaling [51]

1. After forming a new IPv6 address, MN sends a *F-BU* message to oAR, indicating its desire for context transfer.

2. oAR sends a *HI* message to nAR, including the *Smooth Handover Reply (SHREP)* option which is used in an "unsolicited" fashion (*U-SHREP*). The message also includes all the relevant feature contexts and the authentication option.

3. When associating with nAR, MN sends a *Smooth Handover Initiate (SHIN)* message. nAR first verifies the authentication data presented in the *SHIN* message based on the authentication option from oAR. When the authentication passes, nAR activates the transferred context.

4. nAR may include a *SHREP-ACK* option in the *HACK* message being sent to oAR.

This model will also be used in the performance analysis in Chapter 8.

However, for a end-to-end IntServ QoS support, transferring context at the last hop router may be insufficient to completely re-initialize the mobile host's QoS treatment, since some number of additional routers in the path between the mobile host and corresponding node may also need to be involved.

In case that the MN upgrades its QoS request during a handover, the authorization data transferred from oAR to nAR may not be useful to authorize the new QoS request by the nAR.

The integration of "Fast Handovers" and "Context Transfers" aims to achieve seamless handover procedure consisting of the IP connectivity establishment phase and registration phase [52]. The proposal can provide authentication and authorization. Transferred context can be used to set up a SA between MN and the new AR. However, the new SA is identical with the old SA between MN and the old AR. If the old SA is weaken, vulnerability is introduced into the new SA.

Regarding QoS provisioning, DiffServ can be supported by transferring network features from the old AR to the new AR. However, IntServ support can not be provided.

As stated in [57], DoS attacks may be launched from MNs towards the ARs by requesting multiple context transfers and then disappearing. And a bogus AR may flood MNs with packets, attempting DoS attacks, and issue bogus context transfer requests to surrounding routers. Therefore, the risk of DoS attacks exists in "Context Transfer Protocol".

4.3 Summary

How related work can meet the requirements identified in Section 3.4 is summarized in Table 4.2.

In a QoS-aware handover procedure, the condition for a validity of a MN's IPv6 address is different from the handover procedure featuring only mobility. As discussed in Section 4.2.12, the MN's new IPv6 address is valid if the AR performs Duplicate Address Detection (DAD) check successfully or the AR proves the uniqueness of the address. Once the address is validated, the MN is able to send and receive packets using the new address. The operation of binding update does not impact the success of a handover procedure.

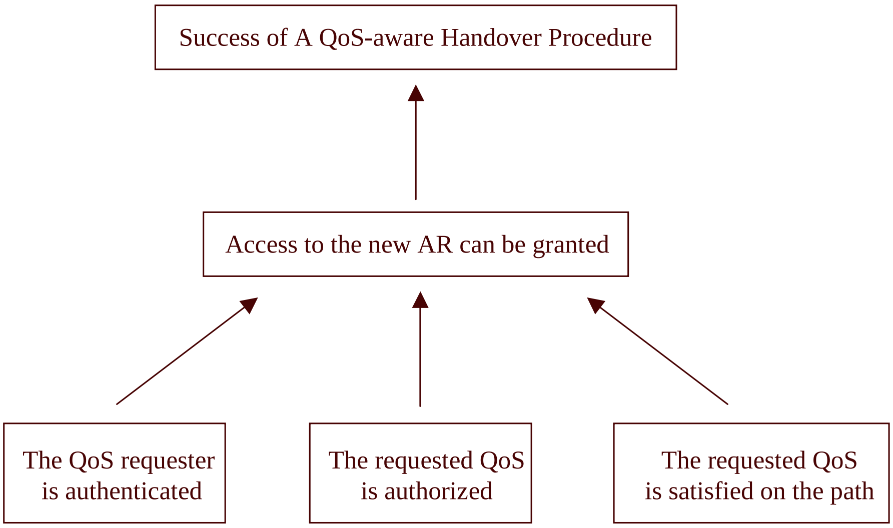

Success of A QoS-aware Handover Procedure
Access to the new AR can be granted
The QoS requester is authenticated
The requested QoS is authorized
The requested QoS is satisfied on the path

Figure 4.16: Factors Determining The Success of a Handover Procedure

In contrast, if a QoS-aware handover procedure occurred in a HMIPv6 based access network, the success of a QoS-aware handover procedure can not be claimed when the addresses (i.e. Local Care-of Address (LCoA) and Global Care-of Address (GCoA))

Requirements	PANA	Diameter MIPv6	token-based	SLA	Mobile RSVP	MIPv6 +RSVP	QoSBU	CASP Client	Moby Dick	FMIPv6 +CT
Authentication	Yes (AAA or CT)	Yes (AAA protocol)	Yes	Yes (AAA)	Yes	Yes	Yes (AAA)	Yes	Yes (AAA)	Yes (token based)
Authorization	Yes (AAA)	Yes (AAA)	Yes (token based)	Yes (AAA)	Not addressed	Not addressed	Yes (AAA)	Yes (token or AAA)	Yes (AAA)	partial (context transfer)
IntServ Support	Not addressed	Not addressed	Yes (hop-by-hop)	No	Yes (RSVP)	Yes (RSVP)	Yes (QoS objects)	Yes (QoS objects)	Yes (QoS broker)	No
DiffServ	Yes (context transfer)	Not addressed	Yes	Yes	No	No	No	No	Yes	Yes
Mobility management	Not addressed	Yes (MIPv6)	Not addressed	Not addressed	Yes (Mobile IP)	Yes (MIPv6)	Yes (HMIPv6)	Yes (Mobile IP)	Yes (MIPv6)	Yes (FMIPv6)
Data protection over wireless	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Low latency	No	Partial	No	Not addressed	Yes	No	Yes	No	Partial	Yes
Avoidance of discolsure of confidential data	Not addressed	Not addressed	Not addressed	No	Not addressed	Not addressed	Not addressed	Not addressed	No	Not addressed
Separation of keys in mobility	Not addressed	No	Not addressed	Not addressed	No	No	No	No	No	No
Protection against DoS	Yes (cookie)	No	Partial	Not Addressed	No	No	No	No	No	No

Table 4.2: Comparison of Related Work

are validated by the AR and the MAP respectively. as shown in Figure 4.16, besides depending on the validity of the new IPv6 addresses, the success of the handover depends on whether *whether the access of a QoS path can granted to the MN*, which further depends on three factors:

1. whether the QoS requested is authenticated
2. whether the requested is authorized
3. whether the requested QoS can be satisfied on the path

Therefore, the *FMIPv6* and *CT* schemes are unable to provide the same efficiency in a QoS-aware handover procedure as in a normal handover procedure featuring only mobility.

Based on the discussion on the design space and the related work, the main design outline is concluded as the following:

- **two step authentication**: a weak authentication is used as the first step in order to protect an access network against DoS attacks and also provide the authentication service; An AAA protocol provides the second step of authentication, as well as authorization.
- **Tight coupling**: the mobility management signaling and QoS signaling can be integrated in one process. Moreover, negotiation for an IPSec Security Association (SA) establishment can be integrated in a registration request.
- **Processing in parallel**: the integrated mobility management and QoS process can happen in parallel with the AAA process.

Chapter 5

Protocol Design and Specification

The design space and related work were discussed in the last chapter. The design outline was also concluded. Based on the conclusion, the details of protocol design and specification for a secure and efficient QoS-aware mobility support in IP-based network will be described in this chapter.

Since mobility design is not the focus within the thesis, the network architecture of Hierarchical Mobile IPv6 (HMIPv6) is used to realize the design ideas. Generally, the design includes the following items:

- **Enhanced advertisements.** The MAP entity in the Hierarchical Mobile IPv6 (HMIPv6) infrastructure advertises its network information and its resource information (e.g. its available bandwidth) periodically at a rate of f_{map}; ARs also advertise their network information and the resource information of the path (e.g. the aggregate bandwidth value) at a rate of f_{ar}. Therefore, in addition to the information of an IPv6 advertisement defined in [64], the advertisements a Mobile Node (MN) receives also include *an MAP option* which contains the MAP's information and *the resource information* which can be provided by the path. Optionally, they also include the address of the "next CASP node" [1] to which a MN sends its registration request.

- **Handover criteria.** An MN has a mobility mode of "eager" or "lazy" based on the fact whether its request of the desired bandwidth has been satisfied by the current path. Its mobility mode is "eager" when its request has been satisfied. In such a case, a MN always prefers to move to a new AR which can satify its desired bandwidth request better. An MN selects a most suited handover target based on the advertised resource information from ARs.

- **Two-step authentication.** In case of Intra Domain Handover (Intra Domain HO),

[1] it means the Cross Application Signaling Protocol (CASP)-enabled the entity in the access network having the short distance to the mobile nodes. AR is assumed to be the "next CASP node" in the thesis.

when receiving a re-registration request, an AR performs a preliminary check of the authenticity of the request as the first-step authentication before performing any further operations on it. If the first-step is successful, the second-step authentication will be performed while other operations are being carried out.

- **Optimization of signaling processes.** In case of Intra Domain Handover (Intra Domain HO), the QoS signaling and mobility signaling processes are integrated in one operation. This integrated signaling process (i.e. QoS+BU process) happens in parallel of the security signaling process (i.e. authorization and the second-step authentication).
- **User data protection by a temporary SA.** In case of Intra Domain Handover (Intra Domain HO), the MN and its associated AR communicates with the protection of a temporary Security Association (SA) if the security signaling process has not been completed when the QoS+BU process is finished.

This chapter is structured as the following: an overview of the network structure and the protocols is given Section 5.1; the design details of the above mentioned items are given in Section 5.2; the protocol operation in case of Intra Domain Handover (Intra Domain HO) which includes all design items is discussed Section 5.3; the corresponding specification of the protocol is illustrated in Section 5.4; a summary is given in 5.5.

5.1 System Overview

Figure 5.1 shows an overview of the network architecture. It is a joint architecture of HMIPv6 and AAA components. A Mobile Node (MN) may move from one access network to another (i.e. inter-domain handover), or within one access network (i.e. intra-domain handover) while keeping a session alive with the Corresponding Node (CN). During a handover, MN's home domain may get involved in security checks and mobility management. The system architecture consists of the following entities:

- **in an access network**,
  - **Gateway (GW):** The gateway is a network node that connects one access network to the Internet. It is also connected with a Mobile Anchor Point (MAP) and the local security authority (i.e. Local AAA Server (AAAL)). In each access network, there is one MAP and one AAAL.
  - **Mobile Anchor Point (MAP):** In Hierarchical Mobile IP, the MAP receives all packets on behalf of MNs it is serving and encapsulates and forwards them directly to the MN's current address (i.e. Local Care-of Address (LCoA)).

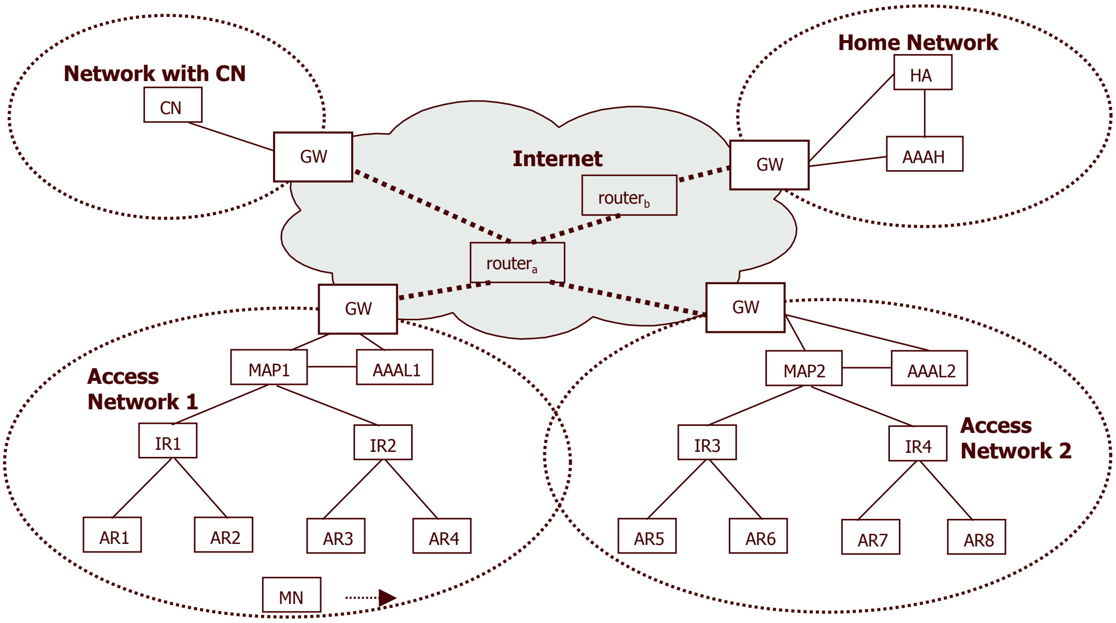


Figure 5.1: An Overview of the Network Architecture

- **Intermediate Router (IR):** Intermediate routers are located between AR and MAP. In a QoS-aware handover procedure, they reserve resources for specific sessions of a MN.
- **Access Router (AR):** The access router is the "last hop" for a MN to be connected with an access network via a wireless channel. It broadcasts advertisements to the mobile users in the radio cells covered by it. [2] In case of intra-domain handovers, when receiving registration requests from mobile users, it performs an authentication check first; when the check passes, it performs operations on the requests. Policies for the DiffServ support are enforced at ARs. Also the wireless channel between ARs and MN should be protected in a proper way.
- **Local AAA Server (AAAL):** AAAL is the local security authority responsible for authentication and authorization checks in the intra-domain cases. It stores the authentication information (e.g. session keys) and the authorization information.
- **Mobile Node (MN):** The mobile node is an IP host with possibly multiple interfaces of different wireless technologies such as $802.11a/b$ for wireless LAN access, W-CDMA (the physical layer of UMTS) for cellular access.

- **in the home network**,

[2] It is assumed that a cellular network in which different radio cells will generally be addressed in different IP subnetworks, each one under the control of one access router.

  - **Home AAA Server (AAAH):** A MN's AAAH is MN's home authority which is able to authenticate the MN. It knows the MN's specific authorization data in a user profile. It also enables MN to obtain a session key to be used for the sessions in the access network based on the long term security relationship with the MN.
  - **Home Agent (HA):** The MN's home agent is responsible for intercepting packets destined to the mobile node's home address, and encapsulating and tunneling them to the mobile node's registered Care-of Address (CoA). In the Hierarchical Mobile IPv6 (HMIPv6) architecture, MN registers its Global Care-of Address (GCoA) to its home agent, which is a globally addressable IPv6 address of the MAP.

5.1.1 Involved Protocols

The scope of the involved protocols within the thesis is Intra Domain Handover (Intra Domain HO). In the power-up cases and the Inter Domain Handover (Inter Domain HO) cases, the registration procedure is assumed to be handled by an appropriate protocol. The Next Steps in Signaling (NSIS) working group of the Internet Engineering Task Force (IETF) is working on the issue of QoS signaling in a generic mobility scenario [30]. To cooperate with the protocols for the Intra Domain Handover (Intra Domain HO) proposed in the thesis, any solutions for the Inter Domain Handover (Inter Domain HO) cases are required to issue a cookie and distribute a set of parameters for a IPSec Security Association (SA) establishment.

The protocol overview of the Intra Domain Handover (Intra Domain HO) cases is shown in Figure 5.2.

- In the infrastructure of AAA protocols, ARs act as the "attendants". The communications between an attendant and AAAL, AAAL and AAAH are realized with the Diameter protocol.

- The CASP Mobility Client protocol is used for QoS signaling. Also the QoS signaling is integrated with the HMIPv6 operations as a combined QoS and mobility process.

- An IPSec SA is used to protect user data over the wireless channel. The IPSec SAs between any pair of MAP, AAAL and ARs are assumed to be pre-established. The trust relationships between AAAL and AAAH, between MN and AAAH are also assumed to pre-exist. The trust relationship refers to Figure 5.3.

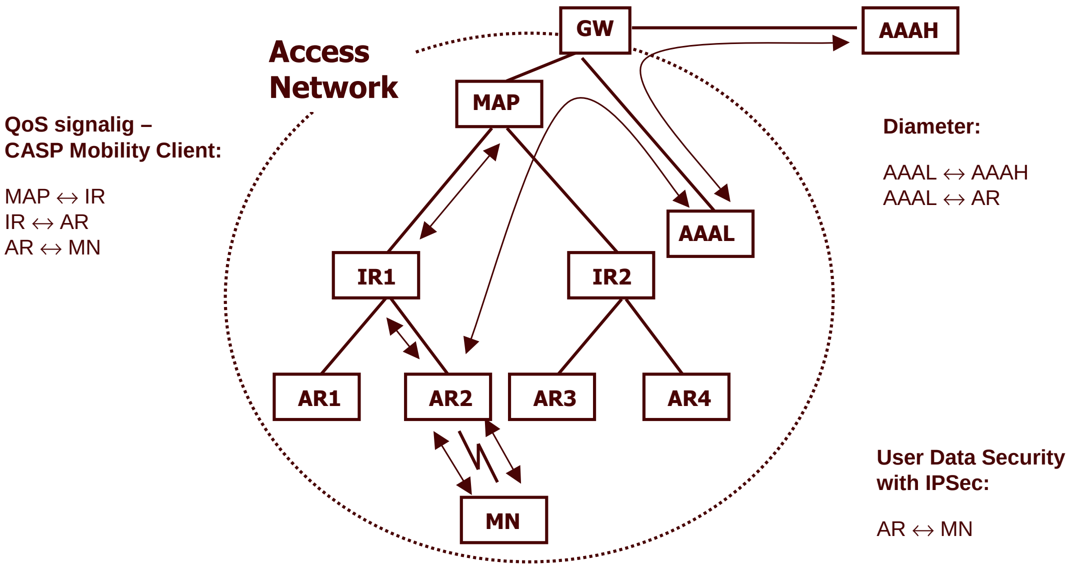


Figure 5.2: A Topological Overview of the Involved Protocols

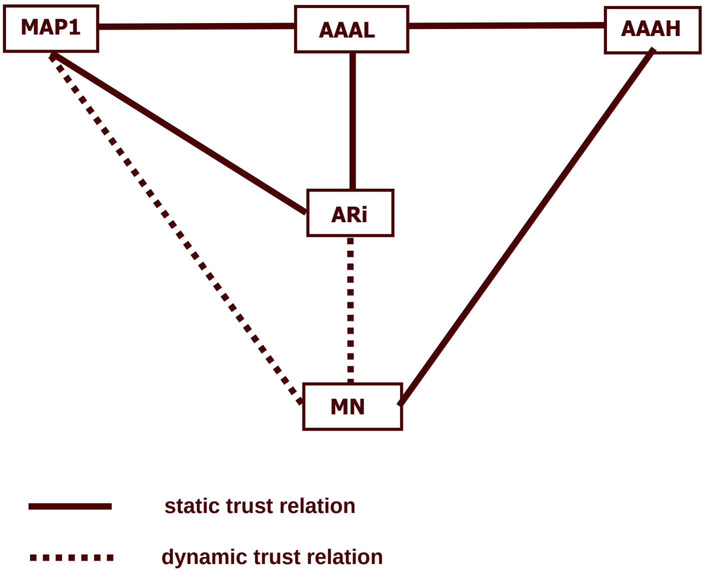


Figure 5.3: Trust Relationships in Intra-Domain Handover Cases

5.1.2 Services Provided by the Protocols

The protocols offer the following services to a Mobile Node (MN):

- *QoS:*
  - Announcement of available resources which can be provided by a certain path via enhanced advertisements;
  - Resource reservation upon QoS requests along a path;

- Updating session states upon upgraded QoS requests on the existing sessions;
- DiffServ support by deploying policies;

- *Mobility:*
  - Registration and re-registration;
  - De-registration in the access network;
  - Support of the “break-before-make” and “make-before-break” handover modes;

- Security
  - Authentication and re-authentication;
  - Authorization and re-authorization;
  - User data protection with IPSec security associations;
  - Signaling protection.

5.1.3 An Overview of A Re-registration Procedure

Since the thesis focuses on re-registration procedures, the remainder of the chapter mainly discusses the design and specifications of the intra-domain handover cases.

An overview of the protocol operations of re-registration procedures in intra-domain handovers is given in Figure 5.4. After receiving enhanced advertisements and deciding to make an handover, MN sends a re-registration request to its “target” AR. The AR first performs an cookie verification as the first step authentication in an two-step authentication approach. If the check is successful, it starts the BU+QoS and re-authorization procedure in parallel. Since the processes go in parallel, it is possible that the the result of QoS+BU process arrives earlier than that of the re-authorization process. In such a case, it is necessary to set up a temporary IPSec SA to protect the user data. From the sending of the re-registration request message and the receipt of the re-registration answer message, CASP mobility protocol is used for the signaling. In the re-authorization process, a specific policy for DiffServ support and the session key for the request need to be transmitted from AAAL to AR. With such information, AR is able to enable DiffServ support and perform the second-step authentication.

5.2 Design Details

In this section the key design items will be described according to the message flow shown in Figure 5.4:

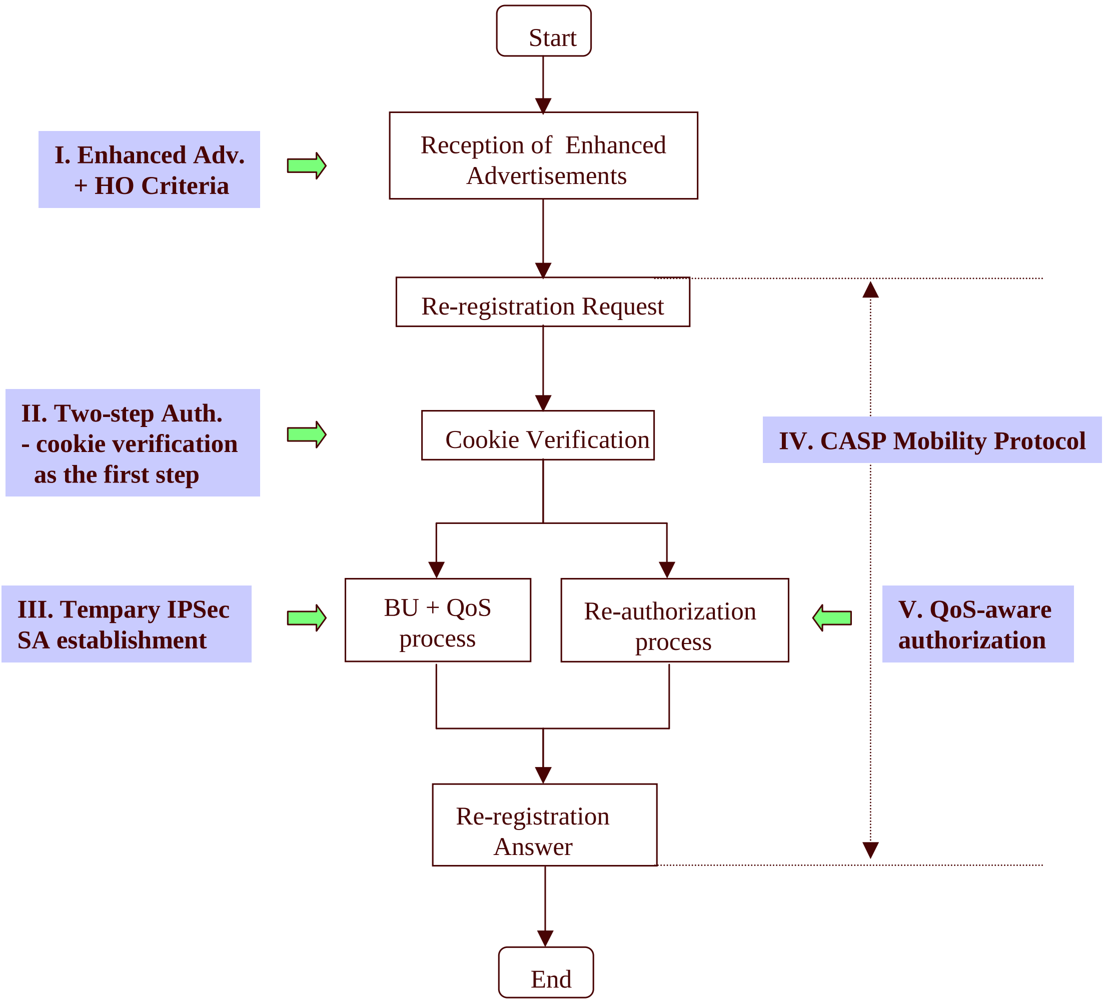

Start
I. Enhanced Adv. + HO Criteria
Reception of Enhanced Advertisements
Re-registration Request
II. Two-step Auth. - cookie verification as the first step
Cookie Verification
IV. CASP Mobility Protocol
III. Tempary IPSec SA establishment
BU + QoS process
Re-authorization process
V. QoS-aware authorization
Re-registration Answer
End

Figure 5.4: An Overview of the Re-registration Procedure and Key Design Items

I. Enhance advertisements and handover decision criteria (Section 5.2.1);

II. a two-step authentication approach with a cookie verification as the first step; (Section 5.2.2);

III. Temporary security association for the user data protection (Section 5.2.3);

IV. CASP Mobility Client Protocol (Section 5.2.4);

V. QoS-aware re-authorization (Section 5.2.5).

5.2.1 Enhanced Advertisements and Handover Decision Criteria

Presently, the ICMP advertisements defined in [64] only include the topology information of the access network. A mobile node can construct a new Care-of-Address (CoA) based on the advertised prefix information. Before performing a handover, a mobile node may receive multiple advertisements from different access routers. The mobile node may select a handover target randomly and send a (re-)registration request. If the path can not satisfy its QoS request, it has to try with another one. It may introduce unnecessary delay to the (re-)registration procedure.

In order to assist mobile nodes to select the most suited access router (i.e. which can satisfy the MN's Desired Bandwidth (DBW)) as the first try, aggregate QoS information of a path is introduced in advertisements. When a mobile node receives such advertisements from different ARs, it is able to select the most suited AR as the handover target upon a handover decision criteria, in which the aggregate QoS information is taken into account.

Advertisement Propagation

MAP is responsible for being discovered by mobile nodes. Therefore, it advertises its presence downstream in the access network with a "MAP option" [68]. Additionally, the MAP puts its available QoS value in the advertisement. If it is a CASP node, MAP will also add its own IPv6 address in the "next CASP node" field.

Every Intermediate Router (IR) or Access Router (AR) is able to receive and propagate advertisement messages from its upstream nodes in the access network. As shown in Figure 5.5, a node in level $n + 1$ sends an advertisement to the nodes in level n, including e.g. its prefix information, the bandwidth value it can provide. A node in level n extracts the useful information from the advertisements being receiving from the node in level $n + 1$, and composes its own advertisement containing

- the prefix information of the upper router and itself;
- the aggregate QoS value the path can provide; (it fill in the QoS value it can provide in the QoS parameter field if the value is smaller than that in the received advertisement from the upper node;
- its own IPv6 address as the address of the "next CASP node" if it is CASP-enabled.

Advertisement Message Format

The message format of an enhanced ICMPv6 router advertisement message is shown in Figure 5.6. The details of IPv6 header and ICMPv6 router advertisement are described in [22] and [64]. The fields of the enhanced features include the IPv6 address of the next CASP node and the Aggregate Available Bandwidth (AABW) information [17].

Next CASP node Option:

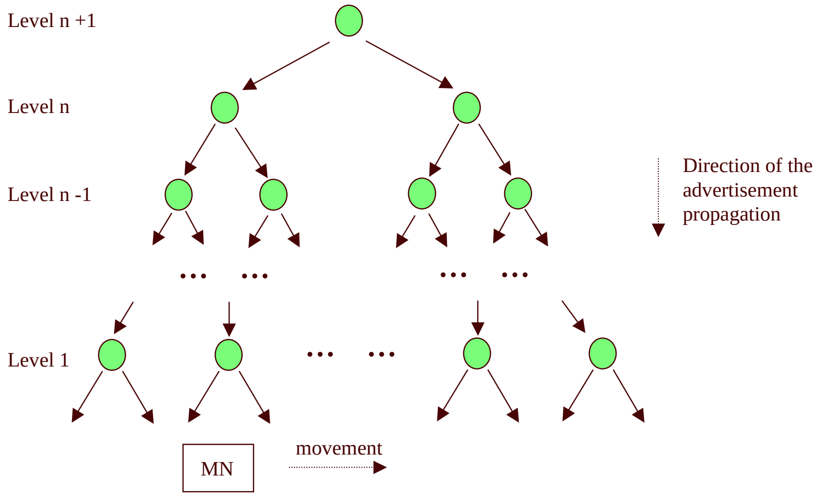


Figure 5.5: Advertisement's Propagation

Type To be Determined (TBD)

Length 8-bit unsigned integer. The length of the option (including the type and length fields) in units of 8 octets. The value is 3.

Reserved This field is unused.

Valid lifetime This value indicates the validity duration of the announced CASP node.

CASP node addr The IPv6 address of "the next CASP node". This address is used as the destination address for the CASP messages which are sent by the mobile node to reserve bandwidth in the access network.

Bandwidth and Price Information Option:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|vers=6 |Prio=15|                Flow Label                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Payload Length        |Next Header=58 | Hop Limit =255|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|      Source Address = router or home agent's address          |
+                                                               +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
| Destination Address = mobile node's address* or All-Nodes     |
+                                        multicast address      +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type=134      | Code=0        |        Checksum               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Cur Hop Limit |M|O|  Reserved |       Router Lifetime         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Reachable Time                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Retrans Timer                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type=3        | Length=4      | Prefix Length |L|A| Reserved 1|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Valid Lifetime                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Preferred Lifetime                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Reserved2                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                Network Prefix Information                     |
+               (Global IPv6 Address of MAP)                    +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type          | Length=3      | Reserved                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Valid Lifetime                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                       CASP node address                       |
+                                                               +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type          |  Length=1     |Flg|       Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Aggregate Available Bandwidth |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5.6: Message Format of An Enhanced Advertisement

Type To be Determined (TBD)

Length 8-bit unsigned integer. The length of the option (including the type and length fields) in units of 8 octets. The value is 1.

Flg A 2-bit flag to indicate the presence of the available bandwidth and the price reference label. 11 means the advertisement contains available bandwidth and price reference label information; 10 means only available bandwidth and no price reference label; 01 means only price reference label and no available bandwidth information.

bandwidth field 16-bit unsigned integer represents the Aggregate Available Bandwidth (AABW) for each mobile node at the router.

The default “next CASP node” is an AR in the thesis. Another application of the area of the “next CASP node” is to carry the address of the bandwidth broker in the access network.

Handover Decision Criteria

In a related work [68] of a QoS-enabled mobility concept based on HMIPv6 , the handover decision was made by a MN based on a “context field” as shown in Figure 5.7. The value of the “context field” is calculated when the MN receives an advertisement from an access router. The flags are arranged in an order (depending on the policy) such that the router with the largest value in the context field is regarded as the target AR for a handover.

This criteria is not suitable when the enhanced advertisement scheme is deployed since the aggregate available bandwidth information is not considered in the handover decision. To take the aggregate available bandwidth information into account for the handover decision, the Aggregate Available Bandwidth (AABW) information is designed to follow the “has MAP-Option”. Therefore, MN’s handover decision criteria in an environment where the enhanced advertisement scheme is deployed is shown in Figure 5.8.

The reasons for the determined position for the new field are given in the following:

- If the AR is located in MN’s home link, the “home link” field is assigned to 1. As MN stays in its home domain, no QoS-aware mobility is necessary. Therefore, the “home link” field should have higher priority than the “AABW” filed.

- In any HMIPv6 architecture, ARs advertise the availability of a MAP. Therefore, the “has MAP option” field must has the value of 1. This field should also have

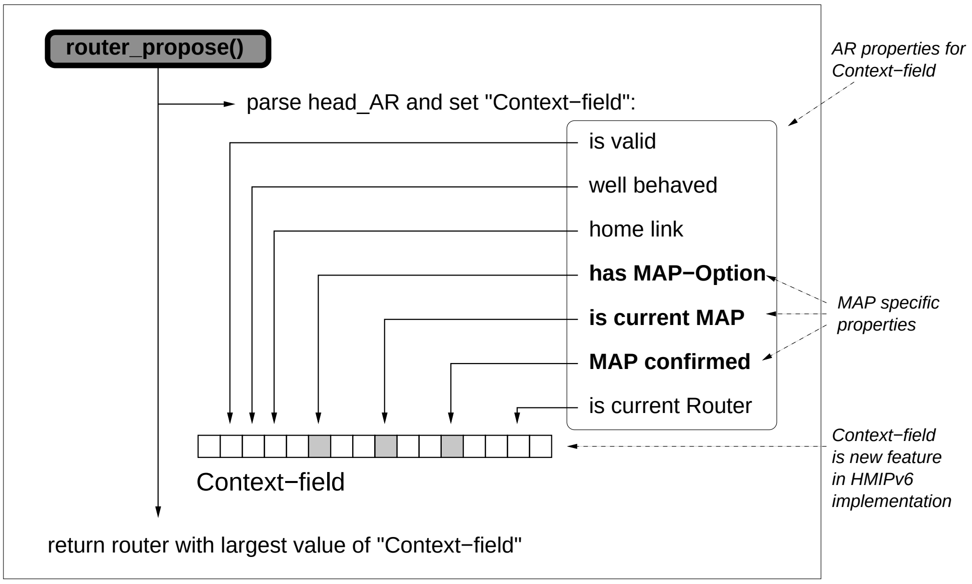

router_propose()
parse head_AR and set "Context−field":
is valid
well behaved
home link
has MAP−Option
is current MAP
MAP confirmed
is current Router
Context−field
return router with largest value of "Context−field"
AR properties for Context−field
MAP specific properties
Context−field is new feature in HMIPv6 implementation

Figure 5.7: Handover Decision Criteria in a Related Work [67]

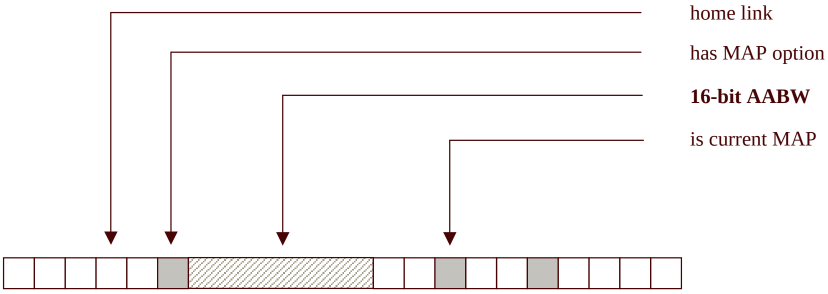


Figure 5.8: Handover Decision Criteria When Enhanced Advertisements Are Present

higher priority.

- A MN may receive multiple “MAP option”s which present different MAPs. Since one access network has only one MAP, MN thus may receive advertisements from different access networks. Even though the focus of the thesis lies in the intra-domain handover cases, It is possible for MN to make a handover to another access network if it can provide better QoS. Therefore, the “is the current MAP” is less important than the “AABW” field. Therefore, the “AABW” field is positioned between “MAP option” and “is current MAP”.

The criteria, however, has a potential problem: as shown in Figure 5.9, it is assumed that both paths can provide e.g. 20 Kbps as the aggregate available bandwidth value. If MN associates with AR1 requesting 5 Kbps successfully, the next advertisement from AR1 will indicate 15 Kbps while AR2 advertises also 20 Kbps. Based on the new criteria, MN moves to AR2 which can provide higher AABW value. As the result, AR2

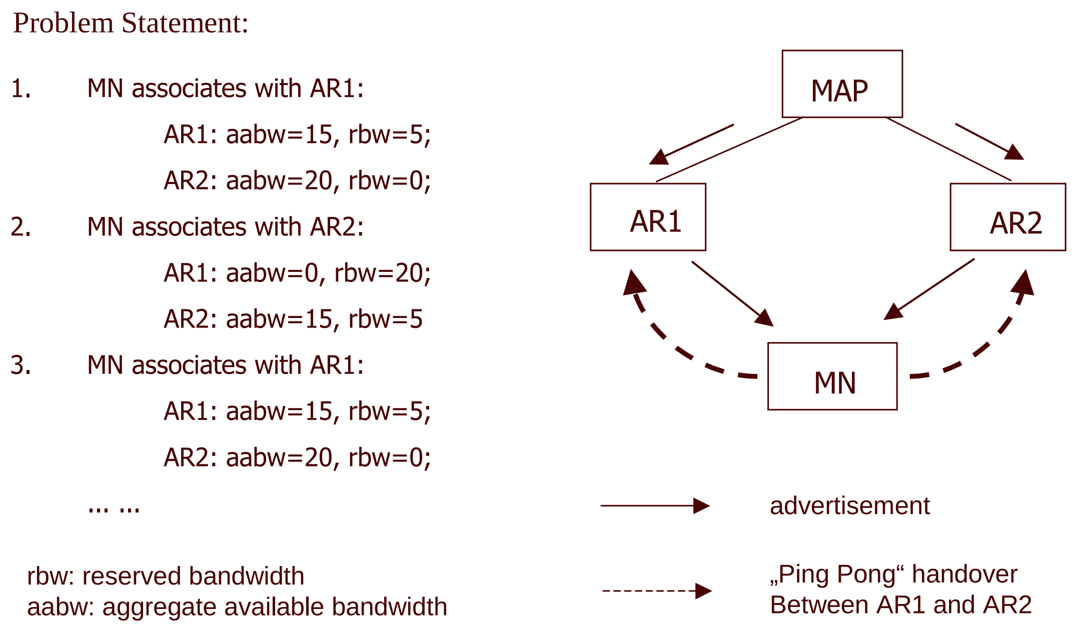


Figure 5.9: The Problem of the Oscillating Handover

advertises 15 Kbps and AR1 is able to provide 20 Kbps. Then MN should move back to AR1. Consequently, MN oscillates between the two ARs.

To solve the problem, two handover mode are defined: **lazy** and **eager**. If MN's DBW is satisfied, MN is "lazy" to move and stays with the current AR even though another AR can provide more bandwidth. If MN's DBW is not satisfied, MN becomes "eager". When MN receives an enhanced advertisement, it computes a context-field value. If the new AR can provide MN more than the MN's current reserved bandwidth, it moves to the new AR; otherwise, it stays with the current AR. The state machine is shown in Figure 5.10.

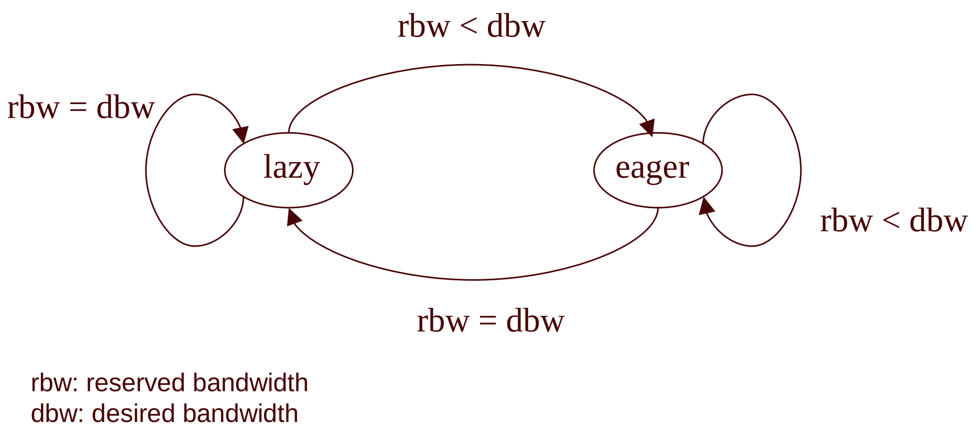


Figure 5.10: The State Machine of MN's Handover Mode

5.2.2 Two-step Authentication

In the two-step authentication scheme proposed in the thesis, a cookie verification serves the first step authentication. In this subsection, the cookie mechanism is isolated from the integrated process and described in details.

The Data Structure of a cookie

Figure 5.11 shows the data fields of a cookie. The purpose of each field may be listed as follows:

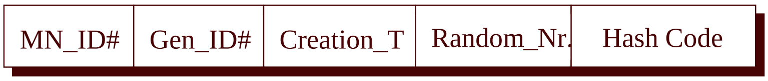


Figure 5.11: Cookie Data Fields

- *MN_ID*: is the MN's unique identifier. This can be a local unique identifier the MN gets after its first registration.
- *Gen_ID*: identifies the cookie generator which is always an AR. It can be the AR's IP address or another unique identifier acceptable in the access network.
- *Creation_T*: is the timestamp marking when the cookie was generated. It is used to limit the cookie's period of validity.
- *Random_Nr*: is used to distinguish two cookies which are generated at the same time.
- *CookieHash*: The hash code is a message digest of the cookie information and a cookie key. The hash function can be of either *HMAC-MD5* or *HMAC-SHA1*. *HMAC-SHA1* is selected in the thesis. The cookie key could be distributed from the MAP to each AR and updated by the MAP periodically. For example a new cookie key is distributed by the MAP every hour or day.

In brief, a cookie is defined according to the following formula:

$$\begin{aligned} CookieInfo \quad &:= \big(MN\_ID, Gen\_ID, Creation\_T, Random\_Nr\big) \\ CookieHash \quad &:= HMAC\big(CookieKey, CookieInfo\big) \\ Cookie \quad &:= \big(CookieInfo, CookieHash\big) \end{aligned}$$

Mechanism Description

- **First cookie generation**

  When a mobile user enters an access network (e.g. it performs a global movement or powers up), the authentication on its first QoS request must involve a trusted network entity (e.g. in the mobile user's home network) because the user is unknown to the access network at the moment.

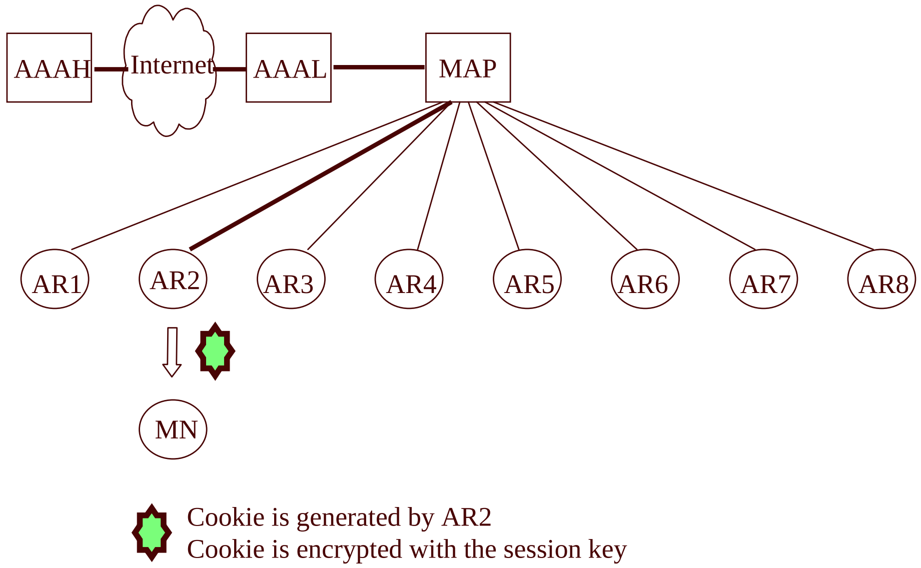

AAAH
Internet
AAAL
MAP
AR1
AR2
AR3
AR4
AR5
AR6
AR7
AR8
MN
Cookie is generated by AR2
Cookie is encrypted with the session key

Figure 5.12: First Cookie Generation

  As shown in Figure 5.12, when an AR receives a QoS request from an MN, it first sends a message to the AAAL server for security check. Since the MN is unknown in the access network, AAAL sends sends message to the MN's AAAH for the authenticaiton check [16]. After the authentication is done successfully at AAAH, the access network knows that the user is credible, AAAL caches the MN's authorization information and the session key which is generated by the AAAH. MAP, AR2 (taken to be the associated AR in the example shown in Figure 5.12) can also learn the session key. MN derives the session key based on its long trust relationship with its home domain.

  AR2 generates a cookie, encrypts the cookie with the session key, inserts the encrypted cookie in the BU acknowledgement (BU ACK) message which is generated by MAP and destined to MN. Thus MN gets its first cookie from the access network.

- **Cookie verification**

  In a local movement, as shown in Figure 5.13, MN presents the cookie to a neighboring AR server (i.e. AR3). Because there is no security association between

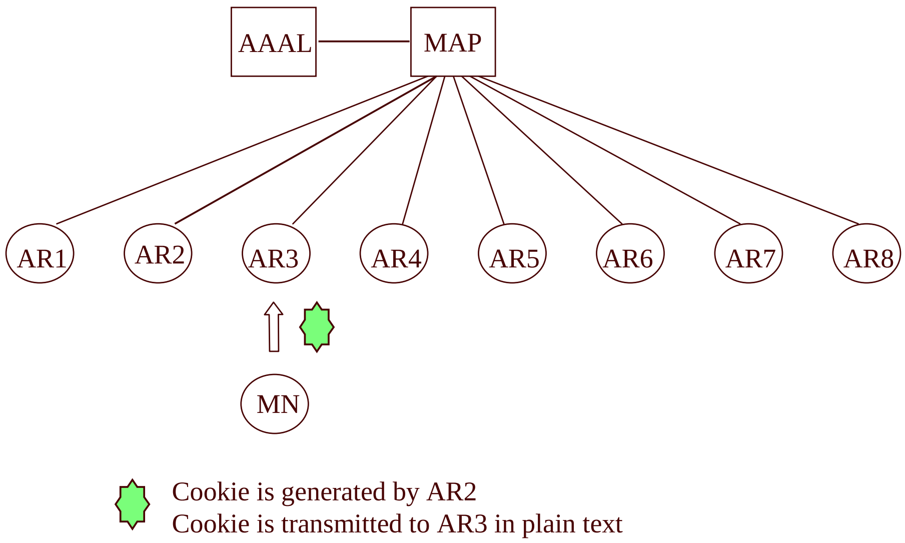

AAAL
MAP
AR1
AR2
AR3
AR4
AR5
AR6
AR7
AR8
MN
Cookie is generated by AR2
Cookie is transmitted to AR3 in plain text

Figure 5.13: AR3 Verifies the Cookie Presented in Plaintext from the MN

MN and AR3 so far, the cookie is transmitted in plaintext. When receiving the cookie, AR3 first verifies that the cookie is valid with the following checks:

- check the timestamp in the cookie to verify the cookie is not expired;
- check the identity of the cookie generator to verify the cookie is created by an AR on its **trusted list**[3];
- verify that the cookie is not on the **notified cookie list**[4];
- if the above checks pass, AR3 computes a key-hashed digest of the cookie information by using a cookie key, and compares the computation result with the hash digest contained in the cookie. If the two hash digests match, the cookie check verification is completed successfully.

After verifying a cookie successfully, AR3 informs AR2 who originally generated the cookie that the cookie has been presented to it. AR2 then notifies all other ARs on its **trusting list**[5] to invalidate the cookie, preventing these ARs to accept it again; indeed, an attacker could intercept the cookie from the open wireless interface and replay it to cheat these ARs for access. The threat will be discussed in Section 6.2.

After the expiration of a cookie's lifetime, all ARs can delete it from the notified cookie list.

[3] An AR's trusted list contains the ARs by which cookies are generated and can be accepted by the AR.

[4] notified cookie list contains all the cookies which are informed by other ARs about the use of the cookies.

[5] An AR's trusting list contains the ARs which are destined to accept the cookies generated by the AR

In case the cookie verification fails, AR3 will drop the registration request silently in order not to devote further resources to this possibly bogus request. If after a certain amount of time the MN has not received any registration answer from the AR, it has to initiate an authentication and authorization process involving AAAL and AAAH as in an inter-domain handover or power-up case, because the old cookie has been transmitted in plaintext and can not be used any longer. Thus the MN can not get the benefit of the optimized handover process.

Even though a verification failure could occur, e.g. due to cookie key updating or interferences from an attacker, the cookie-based preliminary check allows to prevent DoS attacks to the access network without having a major impact on MN's normal operations (i.e. continuous moving without interrupting the service use), as it is reasonable to assume that an MN does not experience these cases very often during its normal operations.

- **New cookie granting**

When the cookie verification is completed successfully, the BU+QoS and re-authorization processes start. The two processes can proceed in series or in parallel. If the two processes are successful, the AR3 can perform the second authentication of the re-registration request when it gets the session key from AAAL. When this check passes, the AR3 generates a new cookie, encrypts it with the session key and inserts it in the BU ACK message as shown in Figure 5.14. The new cookie is used for its next registration and the old cookie is no longer valid in the access network.

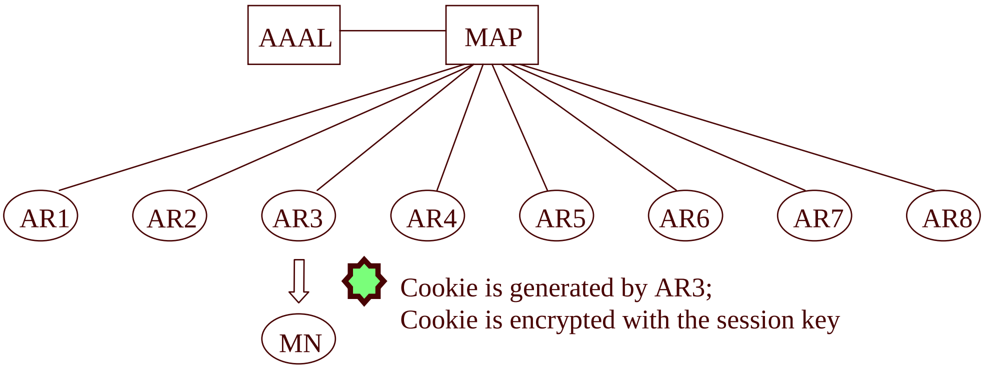

AAAL
MAP
AR1
AR2
AR3
AR4
AR5
AR6
AR7
AR8
MN
Cookie is generated by AR3;
Cookie is encrypted with the session key

Figure 5.14: MN Gets a New Cookie after the Procedure of a Registration

Discussions

There are three main points in the design of this cookie mechanism: the “area of validity”, the notification of a used cookie, and the presentation of a cookie in plain text.

- *Area of Validity:*

  The “area of validity” (AOV) is a group of ARs at which a cookie is valid. In other words, the "area of validity" corresponds to **the trusting list** of the cookie generator.

  When a cookie is presented in plain text to AR3 (see Figure 5.13), if AR3 were not to notify other ARs about the use of the cookie, the cookie could be intercepted by an attacker on air and replayed to other ARs. Consequently, the attacker could gain access at these ARs so as to play DoS attacks on the corresponding paths as shown in Figure 5.15. On the other hand, if each AR were to notify the rest of the access network when receiving a cookie, the propagation of notification messages would generate substantial traffic in the access network.

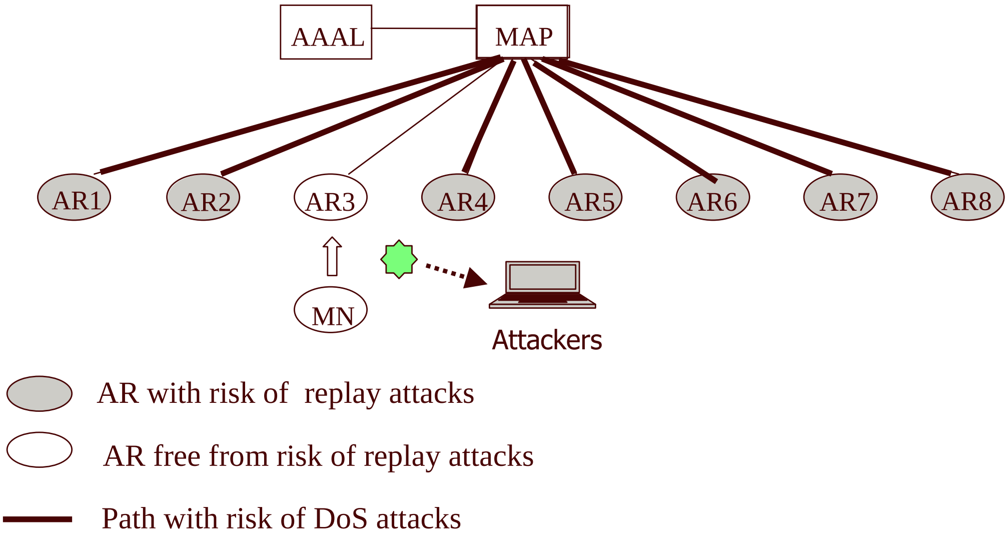

AAAL
MAP
AR1
AR2
AR3
AR4
AR5
AR6
AR7
AR8
MN
Attackers
AR with risk of replay attacks
AR free from risk of replay attacks
Path with risk of DoS attacks

Figure 5.15: Threat of Replayed Cookie without Limited *AOV* and Notification

  Therefore, a limited “**area of validity**” for each cookie is introduced, the nodes in this area being the only ones that can accept the cookie. For example, each AR could have its adjacent ARs only on its **trusting list** and **trusted list**. AR2 would put AR1, AR3 and itself on its **trusting list** to form the “area of validity”, meaning that the cookies generated by AR2 are only accepted by AR1, AR2 and AR3, while being rejected by other ARs (see Figure 5.16). Necessarily, both AR1 and AR3 have AR2 on their own **trusted list** and they can accept cookies generated by AR2.

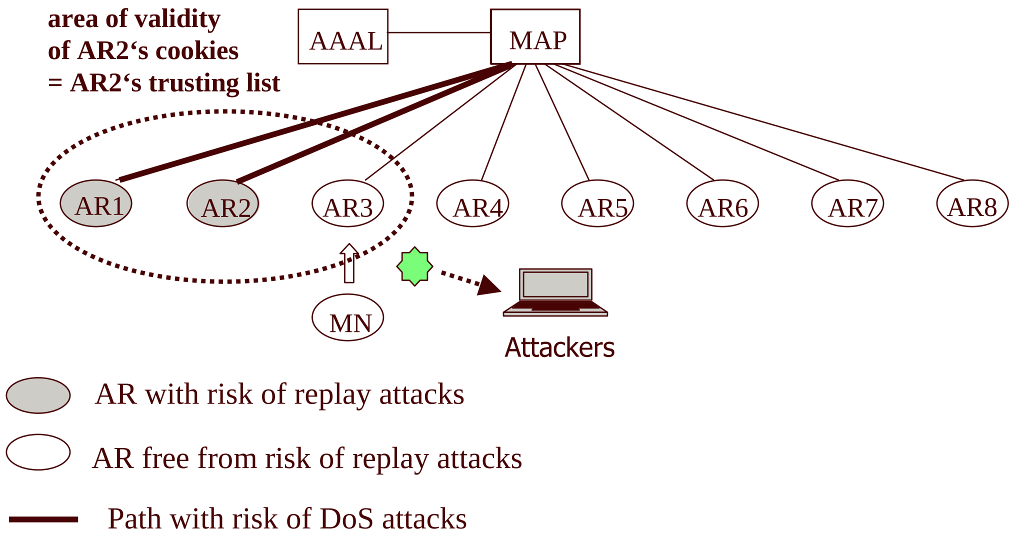


Figure 5.16: Threat of Replayed Cookies with Limited *AOV* and without Notification

- *Notification of used cookies:*

  As shown in Figure 5.17, after verifying the cookie, AR3 notifies immediately AR2 about the cookie use and then AR2 notifies AR1, who is the remaining AR on AR2’s **trusting list**, not to accept the cookie. All ARs are then free from replay attacks, provided that the notification messages propagate faster than the time needed for an attacker to intercept a cookie and replay it.

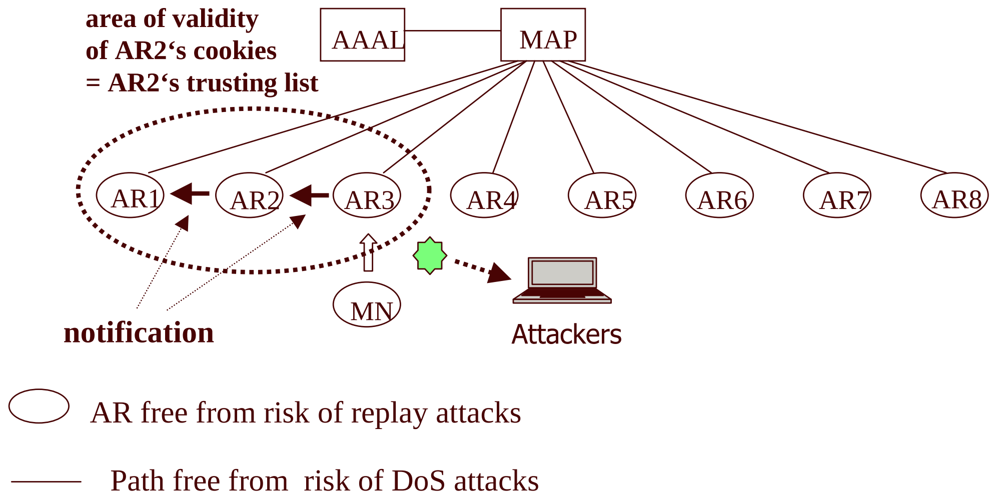


Figure 5.17: Replay Protection with Limited "AOV" and Cookie Notification

- *Presenting a cookie in plain text:*

  Mobile nodes always send cookies in plain text to access routers, as in the case of a handover, since the MN does not yet share a session key with the new access router at the time it sends the cookie. Although the cookie might be intercepted by an attacker when being presented in plain text, the risk of DoS attacks is reduced sufficiently with the mechanisms described above. The reason for this is, that the cookie mechanism reduces the overall number of "credible looking" handover requests, as every cookie can only be presented once.

Summary

In summary, the cookie-based mechanism has the following characterastics:

- Cookies are always generated by an AR. A cookie can be verified by either its generator or other ARs which are destined to accept it.
- A cookie is transmitted encrypted from its generator to the MN and in plain text from the MN to an AR.
- A cookie is used only once since the cookie may be intercepted by an attacker during its transmission on air in plain text.
- The cookie key, which is used for generating and verifying a cookie at ARs, is updated periodically by MAP.
- Each AR maintains two lists: **a trusted list** and **a trusting list**. The AR only accepts the cookies generated by the ARs on the **trusted list**. The cookies generated by an AR can only be accepted by those on the AR's trusting list.
- After accepting the cookie, an AR makes other ARs which can accept the cookie (ARs on the cookie generator's trusting list) know about the use of the cookie by propagating notification messages, in order to prevent further attempts to use the cookie.
- Expired cookies are removed from the notified cookies lists.
- A MN can request the cookie generator to grant a new cookie when its cookie is going to expire.

The performance of the cookie mechanism will be evaluated mathematically in Chapters 6 and 7.

5.2.3 Establishment of a Temporary IPSec SA for User Data Protection

The operations to establish a temporary IPSec Security Association (SA) in case that the re-authorization process is slower than the BU+QoS process is described. The illustration also involves operations of the cookie mechanism.

Generally the data exchanged between a MN and a new AR over the wireless channel are protected with two kinds of security associations:

- **Definitive SA:** A session key $K_{MN,AR}$ used for a definitive SA after a (re-)registration procedure;
- **Temporary SA:** A session key $TK_{MN,AR}$ used for a temporary SA to protect user data before the definitive SA is set up.

The session key $K_{MN,AR}$ is distributed to MN by the new AR under the protection of a session key $K_{MN,AN}$, which is generated by MN's home domain and distributed to the access network under the protection of the long term trust relation ship between AAAH and AAAL. MN can derive the key $K_{MN,AN}$ due to the long term trust relationship with its home domain. The key $K_{MN,AN}$ is used between the mobile node and the Access Network (AN) for authentication and key management purposes.

The hash code of $K_{MN,AR}$ (denoted as $HK_{MN,AR}$) serves the key $TK_{MN,AR}$,

$TK_{MN,AR}$ = $HK_{MN,AR}$

MN performs a hash function on the key $K_{MN,AR}$ to have the key $TK_{MN,AR}$. $TK_{MN,AR}$ is communicated between MN and the new AR with the help of a temporary key distribution key $TmpKeyDistKey_{AN}$, that is distributed by AAAL to each AR in the access network.

Figure 6.1 summarizes the trust model and the keys used in the different SAs in the mechanism.

Mechanism Description

The operations to establish the temporary security association in the proposed intra-domain handovers is shown in Figure 5.19. This message exchanges happen when re-authorization process goes slower than the QoS+BU process.

1. **Registration Answer:** In the registration answer message in a successful inter-domain handover case or a power-up case, the following information is transmitted to MN:

   $r^1_{MN}, E(K_{MN,AN}, (K_{MN,AR_i}, cookie_1, E(TmpKeyDistKey_{AN}, H(K_{MN,AR_i}))))$

   To produce the content, AR1 first generates K_{MN,AR_i} and $cookie_1$. Then it computes the hash code of K_{MN,AR_i} and encrypts it with $E(TmpKeyDistKey_{AN}$. The encrypted product, along with K_{MN,AR_i} and $cookie_1$, is encrypted with $K_{MN,AN}$. Finally the result and a nonce r^1_{MN} are sent to MN. $H(X)$ denotes the computation of a one-way hash function over value X. $E(K, X)$ denotes encryption of X with key K.

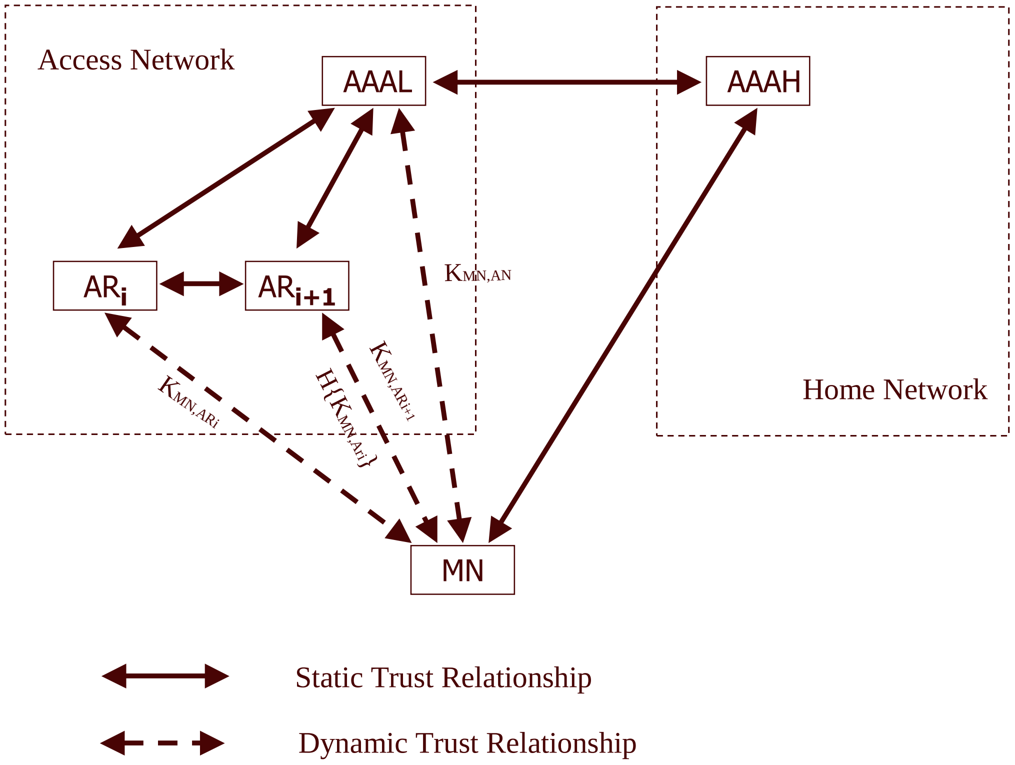


Figure 5.18: Trust Model and the Keys Used in the Different SAs

After verifying the nonce r^1_{MN}, MN can obtain the

$$K_{MN,AR_i}, cookie_1, E(TmpKeyDistKey_{AN}, H(K_{MN,AR_i}))$$

The key

$$K_{MN,ARi}$$

is used to protect the user data between the MN and the old access router AR1.

A temporary session key

$$H(K_{MN,ARi})$$

will be used in an intra-domain handover procedure. $H(K_{MN,AR_i})$ is encrypted with $TmpKeyDistKey_{AN}$ which is unknown to MN.

Furthermore, a cookie

$$cookie_1$$

will be used for a preliminary authentication check during the next intra-domain handover.

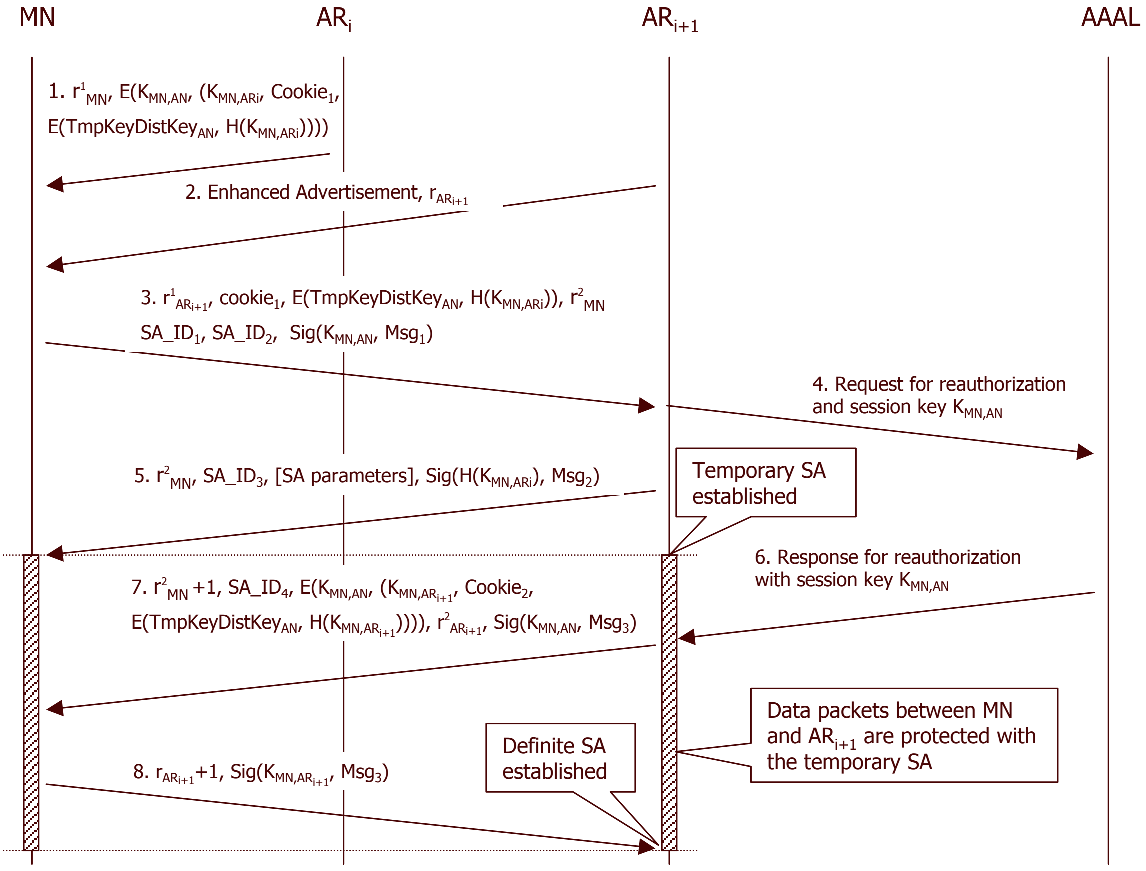

MN
AR_i
AR_{i+1}
AAAL
1. r^1_{MN}, $E(K_{MN,AN}, (K_{MN,ARi}, Cookie_1, E(TmpKeyDistKey_{AN}, H(K_{MN,ARi}))))$
2. Enhanced Advertisement, $r_{AR_{i+1}}$
3. $r^1_{AR_{i+1}}$, $cookie_1$, $E(TmpKeyDistKey_{AN}, H(K_{MN,ARi}))$, r^2_{MN} SA_ID_1, SA_ID_2, $Sig(K_{MN,AN}, Msg_1)$
4. Request for reauthorization and session key $K_{MN,AN}$
5. r^2_{MN}, SA_ID_3, [SA parameters], $Sig(H(K_{MN,ARi}), Msg_2)$
Temporary SA established
6. Response for reauthorization with session key $K_{MN,AN}$
7. $r^2_{MN}+1$, SA_ID_4, $E(K_{MN,AN}, (K_{MN,AR_{i+1}}, Cookie_2, E(TmpKeyDistKey_{AN}, H(K_{MN,AR_{i+1}}))))$, $r^2_{AR_{i+1}}$, $Sig(K_{MN,AN}, Msg_3)$
Data packets between MN and AR_{i+1} are protected with the temporary SA
Definite SA established
8. $r_{AR_{i+1}}+1$, $Sig(K_{MN,AR_{i+1}}, Msg_3)$

Figure 5.19: Message Exchanges of SA Establishment When Re-authorization is Slow

2. **Enhanced Advertisements:** When MN loses its current link, or its mobility mode is "eager", MN needs to make a handover decision upon received enhanced advertisements. The advertisements also contain a random number

$$r^1_{AR_{i+1}},$$

which is used as the challenge in a challenge response mechanism.

3. **Re-registration Request:** When having decided to perform a handover to the new access router AR_{i+1}, the mobile node sends a registration request message which includes

$$r^1_{AR_{i+1}}, cookie_1, E(TmpKeyDistKey_{AN}, H(K_{MN,AR_i})),$$

$$r^2_{MN}, SA\_ID_1, SA\_ID_2, Sig(K_{MN,AN}, Msg_1)$$

The random number

$$r^1_{AR_{i+1}}$$

is obtained from the new access router's advertisement,

$$E(TmpKeyDistKey_{AN}, H(K_{MN,ARi}))$$

from the registration answer message from Step 1, a nonce and two Security Parameters Index (SPI)s

$$r_{MN}^2, SA\_ID_1, andSA\_ID_2$$

are generated by itself. SA_ID_1 and SA_ID_2 are used to indicate the **temporary** IPSec SA and the **definitive** one respectively in the direction from MN to AR_{i+1}.

A $HMAC-SHA1$ hash code is computed with the session key $K_{MN,AN}$ over the the above information and the QoS and BU information as a signature. This signature will be used for the second authentication when AR_{i+1} obtains the session key $K_{MN,AN}$ from AAAL.

All the content will be transmitted as the CASP objects as shown in Figure 5.23.

4. **Request for Re-authorization:** While performing the BU+QoS process, AR_{i+1} sends a request for re-authorization and delivery of the session key $K_{MN,AN}$ to AAAL. The communication between AR_{i+1} and AAAL is protected under the long term trust relationship between them.

5. **Re-registration Answer:** Once the BU+QoS has been finished, while the new access router AR_{i+1} notices that the re-authorization is not complete, it sends the MN the information

   $$r_{MN}^2, SA\_ID_3, [SAParameters], Sig(H(K_{MN,AR_i}), Msg_2)$$

   In addition to MN's nonce r_{MN}^2, AR_{i+1} also includes SA_ID_3 which indicates the SPI for the temporary IPSec Security Association (SA) in the direction from AR_{i+1} to MN, and some necessary IPSec SA parameters such as the cryptographic algorithm. Then AR_{i+1} signs the above information and the BA and QoS information by computing a hash code of it with the key $H(K_{MN,AR_i})$.

   All the content in this step will be transmitted also as the CASP objects as shown in Figure 5.26.

   After the mobile node has checked the nonce r_{MN} and verified the signature, it can resume its session under the protection of the temporary SA.

6. **Answer for the Re-authorization Process:** Along with a successful re-authorization answer from AAAL, AR_{i+1} can also receive the session key

   $$K_{MN,AN}$$

7. **Refresh Request:** After performing the actions as shown in Figure 5.31, AR_{i+1} sends a "Refresh Request" message to MN including

$$r^2_{MN} + 1, SA\_ID_4, E(K_{MN,AN}, (K_{MN,AR_{i+1}}, cookie_2,$$

$$E(TmpKeyDistKey_{AN}, H(K_{MN,AR_{i+1}})))), r^2_{AR_{i+1}}, Sig(K_{MN,AN}, Msg_3)$$

   $r^2_{MN} + 1$ is used as a nonce; SA_ID_4 specifies the SPI for a definitive IPSec SA in the direction from AR_{i+1} to MN.

   $K_{MN,AR_{i+1}}$, $cookie_2$, $E(TmpKeyDistKey_{AN}, H(K_{MN,AR_{i+1}}))$ is information distributed by the new AR AR_{i+1} to MN for the same purpose as in Step 1.

   Since the message requires an acknowledgement message a nonce $r^2_{AR_{i+1}}$ is added in the message to meet the requirement of a challenge response mechanism.

   All the fields are encrypted with the session key $K_{MN,AN}$; all the above information is signed with the session key $K_{MN,AN}$.

8. **Refresh Reply:** After the MN has received this message it checks the nonce and verifies the signature, it is getting ready to use the new IPSec SA and it sends a CASP "refresh reply" message to AR_{i+1} containing

$$r^2_{AR_{i+1}}, Sig(K_{MN,AR_{i+1}}, Msg_4)$$

   $r^2_{AR_{i+1}}$ is used as a nonce and the message is signed with the key $K_{MN,AR_{i+1}}$.

   Figure 5.28 shows the corresponding CASP objects.

Discussion and Summary

The mechanism allows a mobile node to securely exchange data with the new access router during the period from successful establishment of QoS to the completion of the re-authorization and re-authentication checks and establishment of a new definitive SA. During this period, the user data is protected by a temporary IPSec SA with the key $H(K_{MN,ARi})$; After the establishment of the new definitive SA, $K_{MN,AR_{i+1}}$ is used to protect the user data.

By distinguishing the temporary key $H(K_{MN,ARi})$ and the definitive key $K_{MN,AR_{i+1}}$, it ensures that an attacker who somehow comes to know the temporary key can not deduce anything about the definitive session key which is agreed between the MN and new access router due to the fact that the hash function can not be inverted. This is particularly important as otherwise only one key used to protect the user data for a long time would introduce vulnerabilities.

The temporary key $H(K_{MN,ARi})$ can be obtained by MN with the help of the session key $K_{MN,AN}$, and by the new access router AR_{i+1} with the help of the key $E(TmpKeyDistKey_{AN})$.

The messages carry random values which are used in a challenge response mechanism to verify the timeliness of the transmitted information. It is a classical method to prevent the replay attacks. One extension is that a nonce plus 1 is used in a correlated message.

The message exchanges of the crypto aspects of the intra-domain handover cases when the re-authorization process goes faster than the QoS+BU process is shown in Figure 5.20.

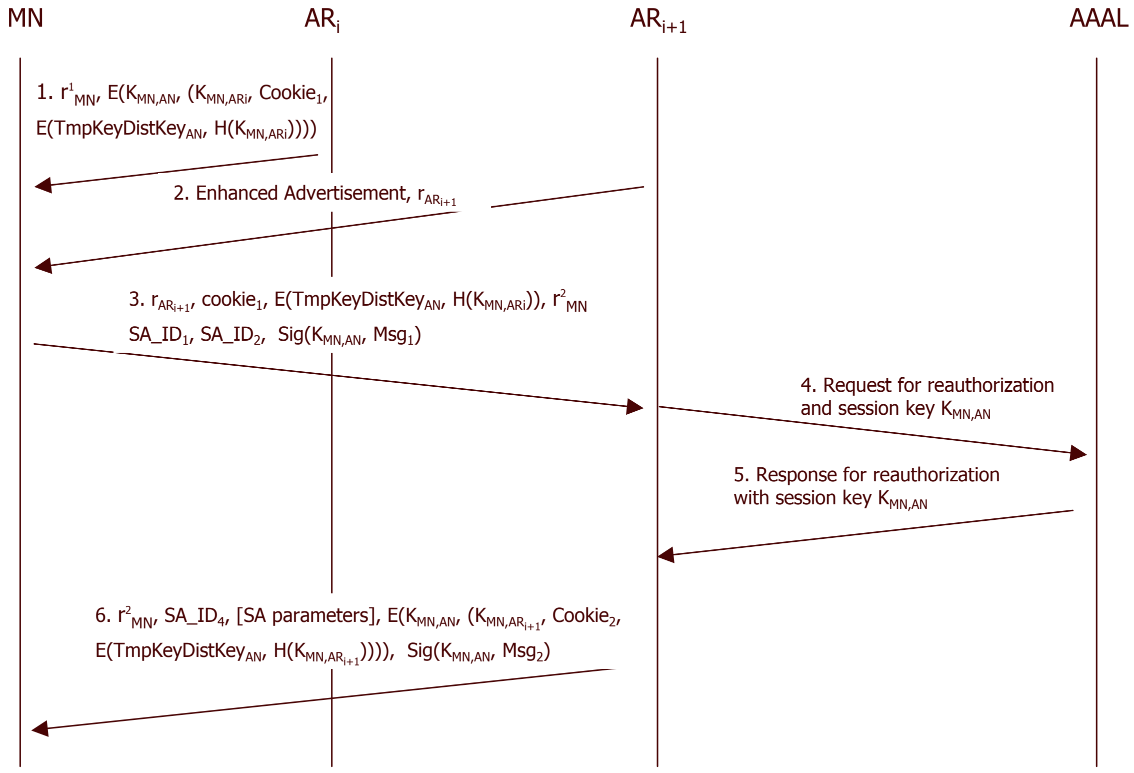

MN
AR_i
AR_{i+1}
AAAL
1. r^1_{MN}, $E(K_{MN,AN}, (K_{MN,ARi}, Cookie_1, E(TmpKeyDistKey_{AN}, H(K_{MN,ARi}))))$
2. Enhanced Advertisement, $r_{AR_{i+1}}$
3. $r_{AR_{i+1}}$, $cookie_1$, $E(TmpKeyDistKey_{AN}, H(K_{MN,ARi}))$, r^2_{MN}
SA_ID_1, SA_ID_2, $Sig(K_{MN,AN}, Msg_1)$
4. Request for reauthorization and session key $K_{MN,AN}$
5. Response for reauthorization with session key $K_{MN,AN}$
6. r^2_{MN}, SA_ID_4, [SA parameters], $E(K_{MN,AN}, (K_{MN,AR_{i+1}}, Cookie_2,$
$E(TmpKeyDistKey_{AN}, H(K_{MN,AR_{i+1}}))))$, $Sig(K_{MN,AN}, Msg_2)$

Figure 5.20: Message Exchanges of SA Establishment When Re-authorization is Fast

When AR_{i+1} receives the acknowledgement message in the BU+QoS process, it notices that the re-authorization process has finished and the corresponding actions have been completed, as Step 6, it sends the message similar to Step 7 in Figure 5.19. The only difference is that r^2_{MN} is sent directly without increasing by 1. When MN receives the message, it resumes the session which is protected by the definitive SA. It is unnecessary for MN to acknowledge the receipt of the message.

5.2.4 CASP Mobility Protocol

This subsection describes the design of the CASP Mobility protocol which is used for the QoS signaling in the re-registration procedure. This protocol is designed based on the CASP QoS Client protocol [85].

The CASP mobility protocol is used between two CASP peers as a end-to-end communication. It is designed to be a modular signaling protocol. Different client layer objects can be appended to one messaging layer and used for different purposes. The general CASP Mobility message format is given in Figure 5.21. The CASP common headers and the CASP client objects are included as the UDP payload. The figure also shows the data formats of messaging layer common header, client layer common header and an object. The length of a field is shown in bytes. The fields are explained in Appendix.

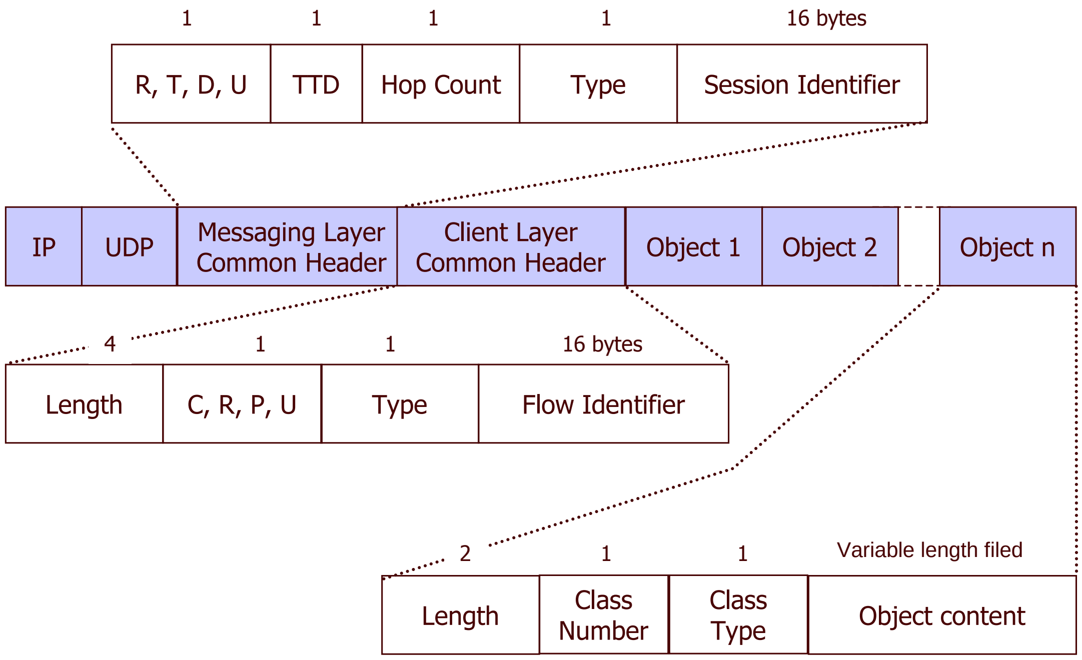


Figure 5.21: CASP Mobility Client PDU Structure

The End to End Communication

In the intra-domain handover procedure, the BU signaling is “piggybacked” by the QoS signaling. A *BU object* is included as one of the CASP Client objects. As shown in Figure 5.22, each the QoS signaling step is a end-to-end communication. The *Re-registration Request* and *Re-registration Answer* messages are communicated between MN and AR. The *CASP QUERY* message propagates from one CASP node to another (as an end-to-end communication) until it hits the Cross-over Router (CR);Similarly, the *CASP RESERVE* propagates from the CR to the new AR which is assumed to be a CASP node; the *CASP TEARDOWN* propagates from the CR to the old AR which is also assumed to be a CASP node with the same manner.

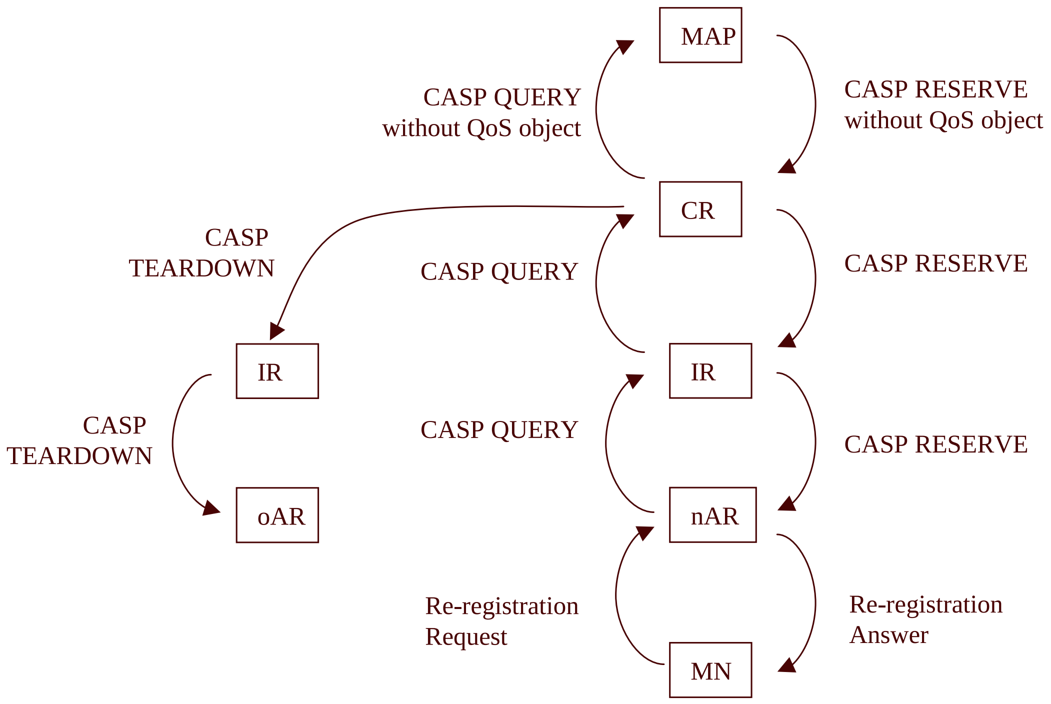


Figure 5.22: End to End Communication for the QoS Signaling

The *CASP QUERY* message is used to reserve resources along a path. If the route is symmetric (i.e. upstream and downstream traffics use the same path), the message enables bidirectional reservation. If the route is asymmetric, the message reserves resources for either upstream or downstream. The reservation for the other direction can be done by sending another CASP RESERVE message without the BU information. The symmetric case is shown in Figure 5.22. The *CASP RESERVE* message is used to confirm the previous reservation for a session; The *CASP TEARDOWN* message is used to release the old QoS path. When the CR is not the MAP, it communicates CASP with MAP containing only a BU or BA object for the purpose of binding update.

Modular Structure

The CASP Mobility Client protocol has the modular structure which is one of the features of the CASP protocol [86]. The necessary information are carried as objects in the client layer. CASP messaging layer header, CASP client layer header and client layer objects are UDP payload.

- **Re-registration Request:**

  The packet structure of *Re-registration Request* which is Step 3 in Figure 5.19 is shown in Figure 5.23.

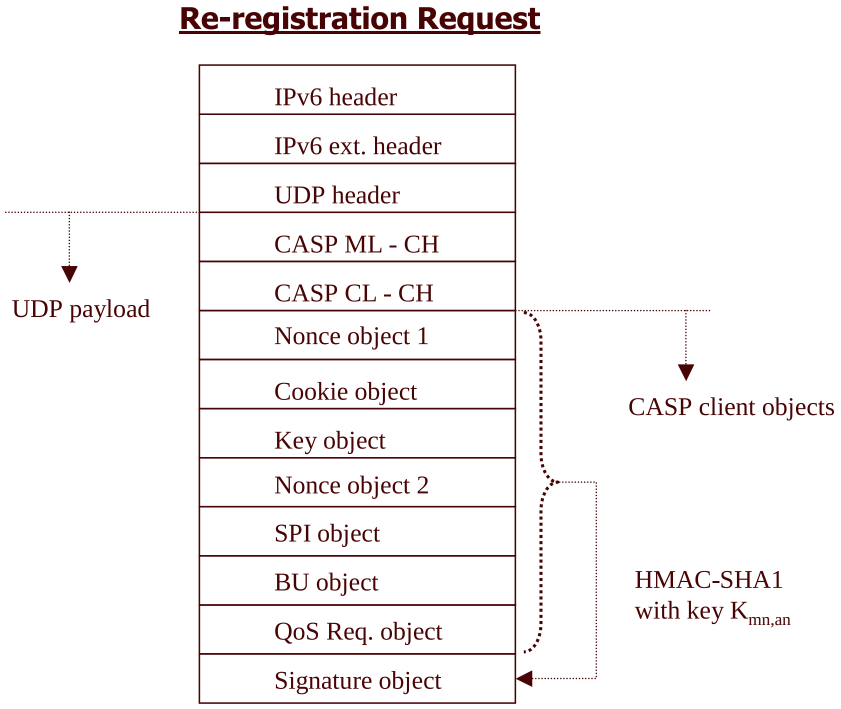


Figure 5.23: Packet Structure of *Re-registration Request*

Nonce object 1 is $r_{AR_{i+1}}$ whose data format is given in Figure A.6. In the thesis the nonce size is designed to be 80 bits in the cases of *HMAC-SHA1* since it is assumed that the two communication peers trust the nonce randomness of both sides. In such a case, the nonce size is the half of the key size;
Key object is $E(TmpKeyDistKey_{AN}, H(K_{MN,ARi}))$ whose data format is given in Figure A.8;
Nonce object 2 is r^2_{MN};

- **CASP QUERY:**

  The packet structure of *CASP QUERY* which is sent to from one CASP node to its next CASP node is shown in Figure 5.24.

  The *BU object* is used for the mobility management operation which is processed only by MAP;

  The *QoS Req. object* carries the QoS requirement of a MN's session and is examined by each CASP node along the path.

- **CASP RESERVE:**

  The packet structure of *CASP RESERVE* which is used to confirm the reservation along the path is shown in Figure 5.25.

  The *BA object* contains the binding acknowledgement;

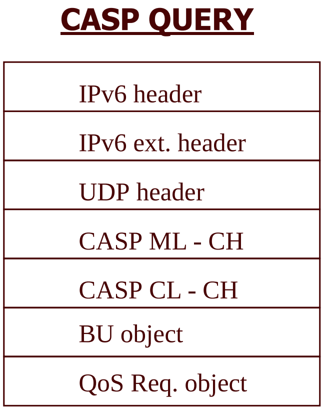


Figure 5.24: Packet Structure of *CASP QUERY*

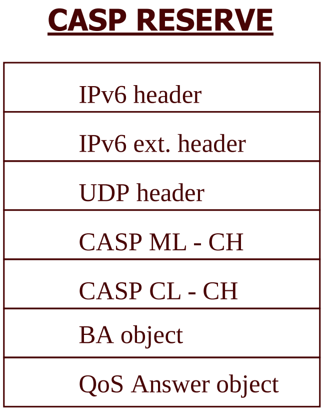


Figure 5.25: Packet Structure of *CASP RESERVE*

The *QoS Answer object* is examined by each CASP node along the path.

- **Re-registration Answer:**

  The packet structure of *Re-registration Answer* which is Step 5 in Figure 5.19 is shown in Figure 5.26.

  The *Nonce object* is r^2_{MN};
  the *Security object* contains the parameters for the IPSec SA set up such as the algorithm;
  the *SPI object* holds the SA_ID_3;

- **REFRESH REQ:**

  The packet structure of *REFRESH REQ* which is Step 7 in Figure 5.19 is shown in Figure 5.27.

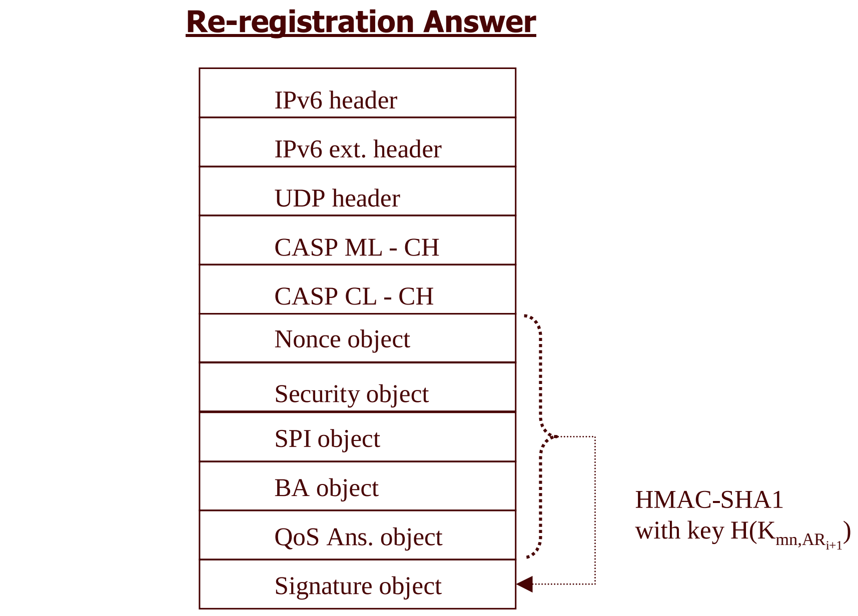


Figure 5.26: Packet Structure of *Re-registration Answer*

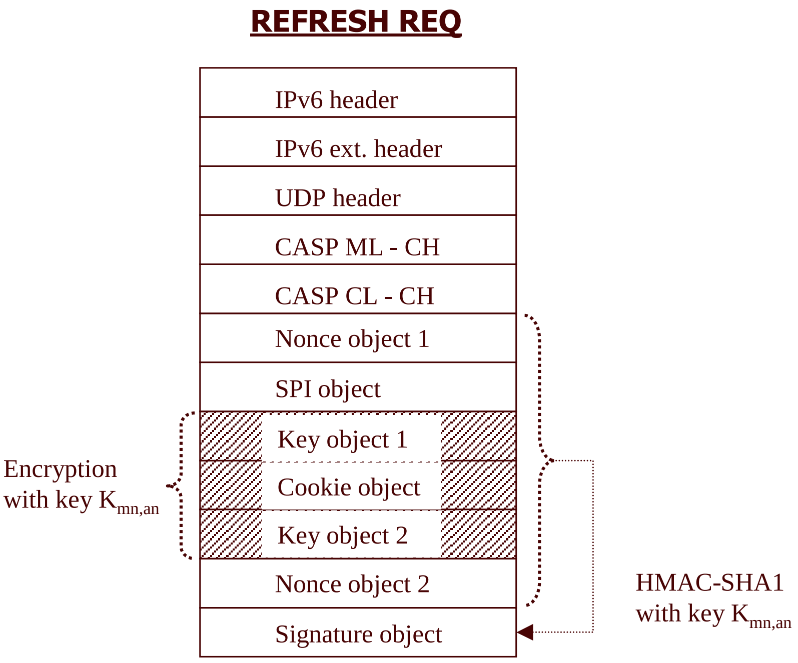


Figure 5.27: Packet Structure of *REFRESH REQ*

The “Nonce object 1” is $r_{MN}^2 + 1$;
the “SPI object” holds the SA_ID_4;
the “Key object 1” contains $K_{MN,AR_{i+1}}$;
the “Key object 2” holds $E(TmpKeyDistKey_{AN}, H(K_{MN,AR_{i+1}}))$;
the “Nonce object 2” is $r_{AR_{i+1}}^2$.

Note that the “Nonce object 1”, “Cookie object” and “Nonce object 2” are encrypted with key $K_{MN,AN}$.

- **REFRESH REPLY:**

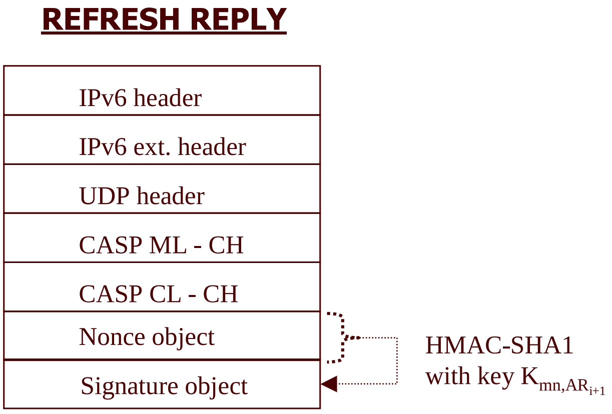


Figure 5.28: Packet Structure of *REFRESH REPLY*

The packet structure of *REFRESH REPLY* which is Step 8 in Figure 5.19 is shown in Figure 5.28.

The “Nonce object” is $r^2_{AR_{i+1}}$.

The data formats of the involved CASP client objects are given in Appendix.

Message Flows of CASP Mobility Client Protocol

Figure 5.29 illustrates the the message flows of CASP Mobility Client Protocol.

The main operations are listed as follows:

1. MN: When MN decides to make a handover based on the handover criteria after receiving an enhanced advertisement, it composes a Re-registration Request message, addressing to "the next CASP node" which is assumed to be the nAR in the thesis;

2. AR: AR performs the first authentication check by verifying the presented cookie.

3. AR: if AR does not know the session ID, it checks resource according to QoS information in the BU object: if it can satisfy Desired Bandwidth (DBW), it reserves and composes a CASP QUERY message to its next CASP node; if not but it can satisfy Acceptable Bandwidth (ABW), it reserves what it can provide, composes a CASP QUERY message with modified DBW, and sends it its next CASP node; if not, it replies a FAILURE message to MN.

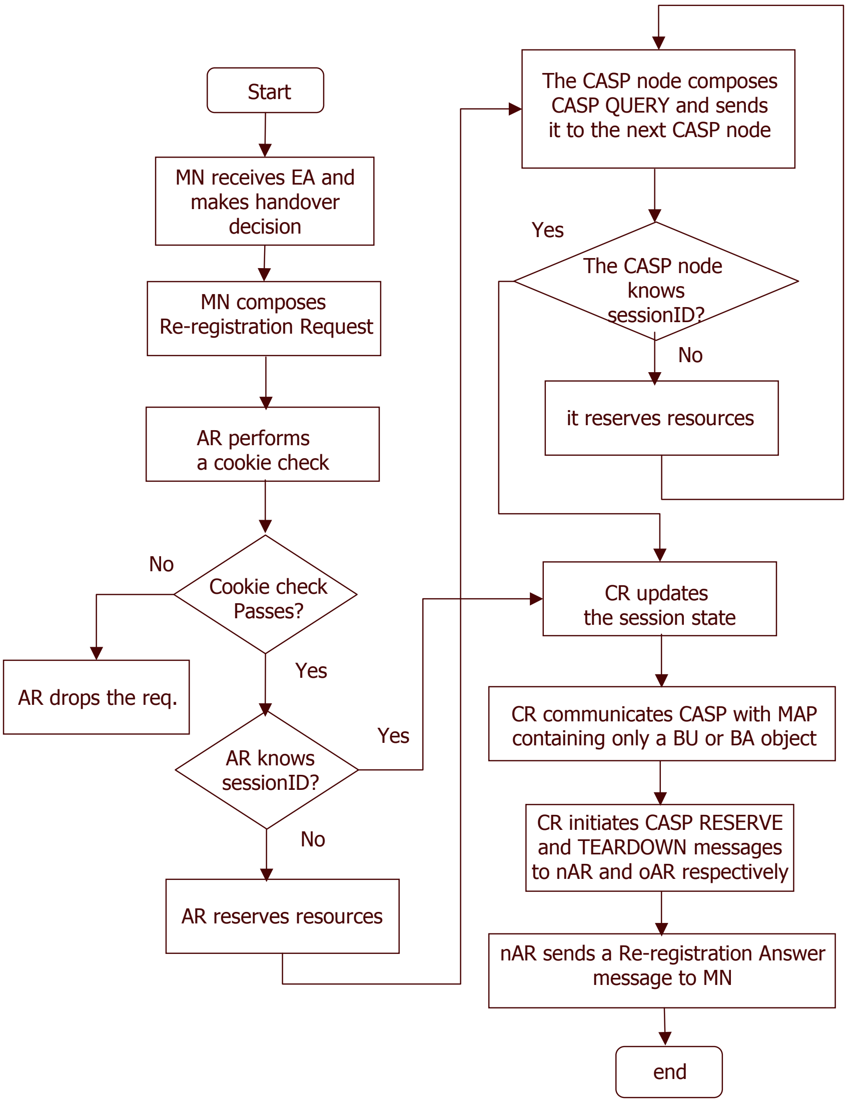

Start
MN receives EA and makes handover decision
MN composes Re-registration Request
AR performs a cookie check
Cookie check Passes?
No
AR drops the req.
Yes
AR knows sessionID?
Yes
No
AR reserves resources
The CASP node composes CASP QUERY and sends it to the next CASP node
Yes
The CASP node knows sessionID?
No
it reserves resources
CR updates the session state
CR communicates CASP with MAP containing only a BU or BA object
CR initiates CASP RESERVE and TEARDOWN messages to nAR and oAR respectively
nAR sends a Re-registration Answer message to MN
end

Figure 5.29: The Basic Operations of CASP Mobility Client Protocol

4. Step 3 repeats until the Cross-over Router (CR) is located: when a IR knows the session ID in the CASP message, it determines itself to be the CR. Then it updates the session state such as the its “next CASP node” towards MN for the session.

5. The CR communicates with MAP by including only a BU or BA object in a CASP message for the purpose of binding update. When the CASP RESERVE message containing a BA object from MAP arrives, the CR inserts a QoS Ans object in the CASP RESERVE message and sends it towards the new AR; meanwhile it initiates a TEARDOWN message towards the old AR respectively.

6. The new AR replies a Re-registration Answer message to MN.

Discussion

Although the Figure 5.31 shows the case where MAP is the CR, CR unnecessarily locates at MAP. The Figure 5.29 presents the generic operations of the CASP Mobility protocol.

In comparison with the related work - CASP QoS Client protocol which requires 8 or 10 message exchanges (see Figure 4.10) before a CASP request message can be processed, the proposed CASP Mobility protocol needs much less exchanges to perform a re-registration.

In the CASP Mobility protocol design, a modular structure is used for the cookie exchange, IPSec SA establishment, QoS path set up and mobility management.

5.2.5 QoS-aware Authorization

Since the re-authorization process in intra-domain handover cases relies on the result of the authorization process in the power-up or inter-domain handover case, the authorization process in the power-up or inter-domain handover case is described before the re-authorization process in intra-domain handover cases. Bandwidth is used as the single QoS parameter in the QoS-aware authorization process.

In Inter-domain handover

Since during the inter-domain handover the mobile node is unknown to the visited access network, AA checks must be performed before the resource reservation process. Therefore, Diameter Mobile IPv6 process [25] takes place as the first round trip between the mobile node and its home domain.

After the process, the authentication and authorization data (e.g. admission control policy, entitled authorization data) is available in the access network, the security association is set up. The mobile node obtains a token (i.e. a cookie) for further QoS requests.

Note that MN may include a selected value or range of bandwidth for the authorization. Correspondingly, it will obtain a token or cookie corresponding to the selected value. The policy or authorization data used for re-authorization locally is in accordance with the selected value.

As the second round trip between the mobile node and the corresponding node, QoS signaling proceeds by using CASP QoS Client Protocol.

As a result of the inter-domain handover procedure, AAAL caches the authorized bandwidth value and policy, the flow state information such as session key $K_{MN,AN}$, session ID and reserved bandwidth for the session. Caching the information may benefit the intra-domain handover in terms of low-latency since AAAL can authorize a QoS request without involving AAAH. It is noted that the subscribed bandwidth value is not disclosed to the visited network if the selected value used for authorization is less than the the subscribed value.

The policy for the DiffServ support has been deployed in the associated AR. The AR has established a security association with the mobile node. The mobile node gets a cookie which the next re-registration procedure in the access network.

When MN leaves the visited network without notification or it powers down unintentionally, AAAL updates the session information based on its lifetime.

In Intra-domain handover

In the intra-domain handovers, when the cookie verification passes, the new AR initiates the BU+QoS and the re-authorization process simultaneously. Figure 5.30 shows the message flow of the re-authorization process during the intra-domain handover scenarios.

When receiving a re-authorization request from an AR, AAAL compares the received DBW with the corresponding Cached Bandwidth Value at AAAL (CBW):

- *CBW > DBW:* the DBW is authorized by AAAL. AAAL sends a positive re-authorization result, the session key $K_{MN,AN}$ and the policy for the DiffServ support to the AR. Then it starts updates billing;
- *CBW < DBW:* AAAL sends a request including the DBW to AAAH for re-authorization. AAAH performs the same authorization process as in power-up or inter-domain handover cases. When receiving the result from AAAH, AAAL takes actions according to the result:
  - in case that DBW is fully authorized, AAAL updates its cached bandwidth value with the new DBW. It then sends a positive re-authorization result, the session key $K_{MN,AN}$ and the policy for the DiffServ support to the AR. In the positive re-authorization ACK message, it indicates that the new DBW is authorized. Finally it starts updates billing;
  - in case that SBW is distributed. AAAL learns that the new DBW is not authorized, instead, the SBW which is smaller that the new DBW is authorized. Then AAAL updates the cached bandwidth value with the SBW. It then sends a positive re-authorization result, the session key $K_{MN,AN}$ and the policy for the DiffServ support to the AR. In the positive re-authorization, it needs to indicate that only the SBW is allowed to by used by the MN. Finally it starts updates billing;
  - in case of a negative re-authorization result is received from AAAH, AAAL forwards the negative result to the AR.

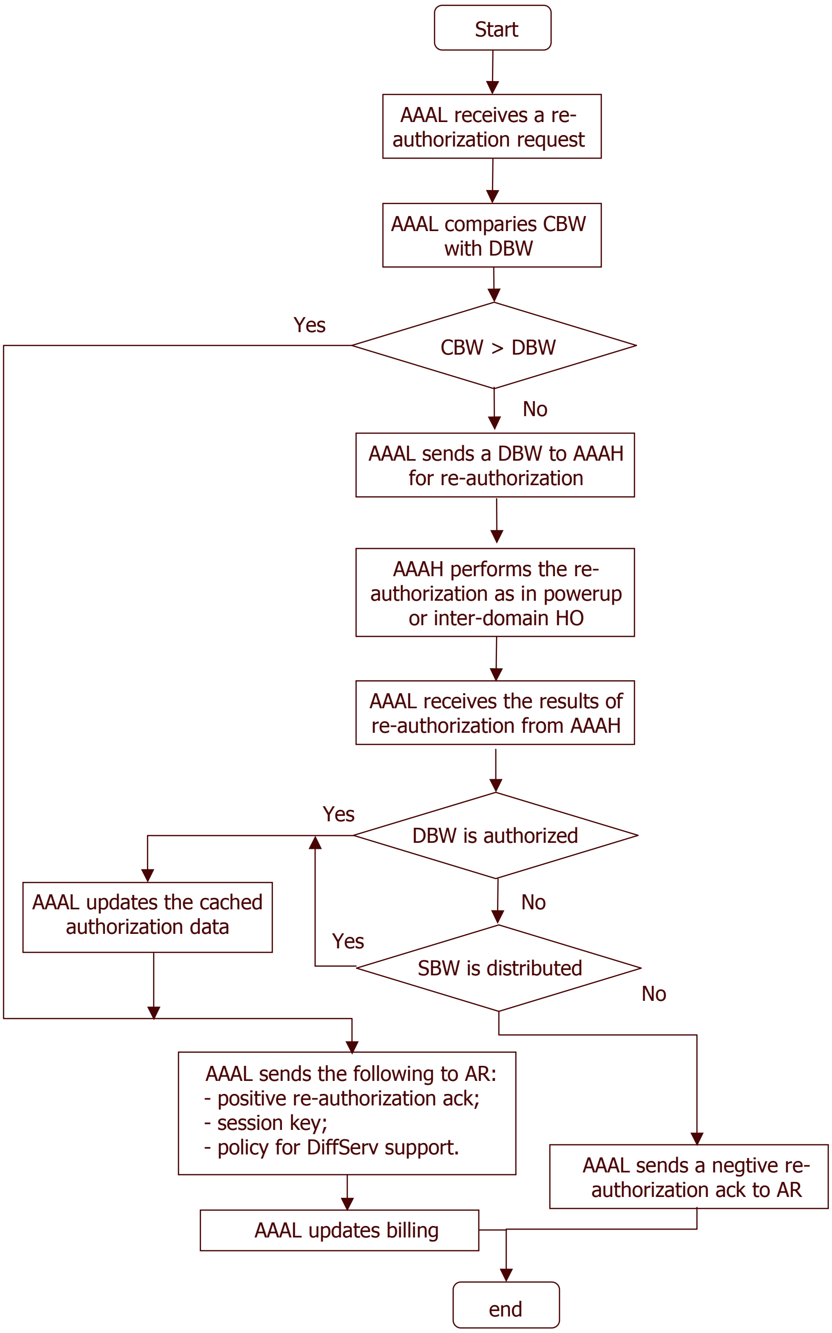

Start
AAAL receives a re-authorization request
AAAL companies CBW with DBW
CBW > DBW
Yes
No
AAAL sends a DBW to AAAH for re-authorization
AAAH performs the re-authorization as in powerup or inter-domain HO
AAAL receives the results of re-authorization from AAAH
DBW is authorized
Yes
No
AAAL updates the cached authorization data
SBW is distributed
Yes
No
AAAL sends the following to AR:
- positive re-authorization ack;
- session key;
- policy for DiffServ support.
AAAL sends a negtive re-authorization ack to AR
AAAL updates billing
end

Figure 5.30: The Message Flow of the Re-authorization Process

Discussion and Summary

If the MN starts new flows or it upgrades its requested QoS during a handover, it is possible that the local network is not capable to perform the re-authorization based on the cached authorization data. In such a case, the AAAH has to get involved. Therefore, it is most likely that the result of the re-authorization process arrives at the new AR later than that of the BU+QoS process. In such a case,

- when the positive re-authorization result contains the SBW, AR needs to check whether the reserved bandwidth in the new path is greater than the SBW (i.g. the bandwidth value the MN is authorized to use). If yes, the new AR sends a session refresh request containing a new CASP Mobility QoS Request Object with the SBW as the desired bandwidth value;

- when a negative re-authorization result arrives, the new AR needs to sends a CASP Mobility TEARDOWN message to remove the resource reservation along the path.

In order to reduce the possibility to involve AAAH for re-authorization, MN may use a selected bandwidth value for the first authorization in case of power-up or inter-domain handover cases. The selected value is what the MN anticipates to use during its stay in the access network. It is independent from the desired bandwidth in the first QoS request.

In summary, the proposed QoS-aware authorization process has the following properties:

- a range of QoS parameter rather than a specific value is used to simplify the authorization negotiation process;
- in order to prevent the subscribed value of a QoS parameter from being exposed unnecessarily in the visited access network, AAAL caches an authorized data is cached at AAAL for the purpose of re-authorization;
- in case of re-authorization, when AAAL is not cable for the task, AAAH is involved. The cached authorization data may be updated if necessary;
- AAAL is enabled to monitor the overall resource utilization by an MN, it can prevent MN's misbehavior to use more resources than it is entitled to. It is also responsible for billing and charging;
- As one of the results of a successful (re-)authorization process, MN's specific policy can be deployed at the new AR from AAAL to support DiffServ.

In the following section, the protocol operations in a complete re-registration procedure will be described including all design items.

5.3 Protocol Operations in Re-registration Procedures

A secure, efficient and QoS-aware mobility support is important in high mobility scenarios. In this section, the protocol operations in re-registration procedures are described by a time-line diagram.

As shown in Figure 5.4, after the cookie verification passes, the BU+QoS process and re-authorization process happen in parallel. The protocol operations are described in two cases: re-authorization process is slower or faster than BU+QoS process.

5.3.1 Re-authorization Process is Slow

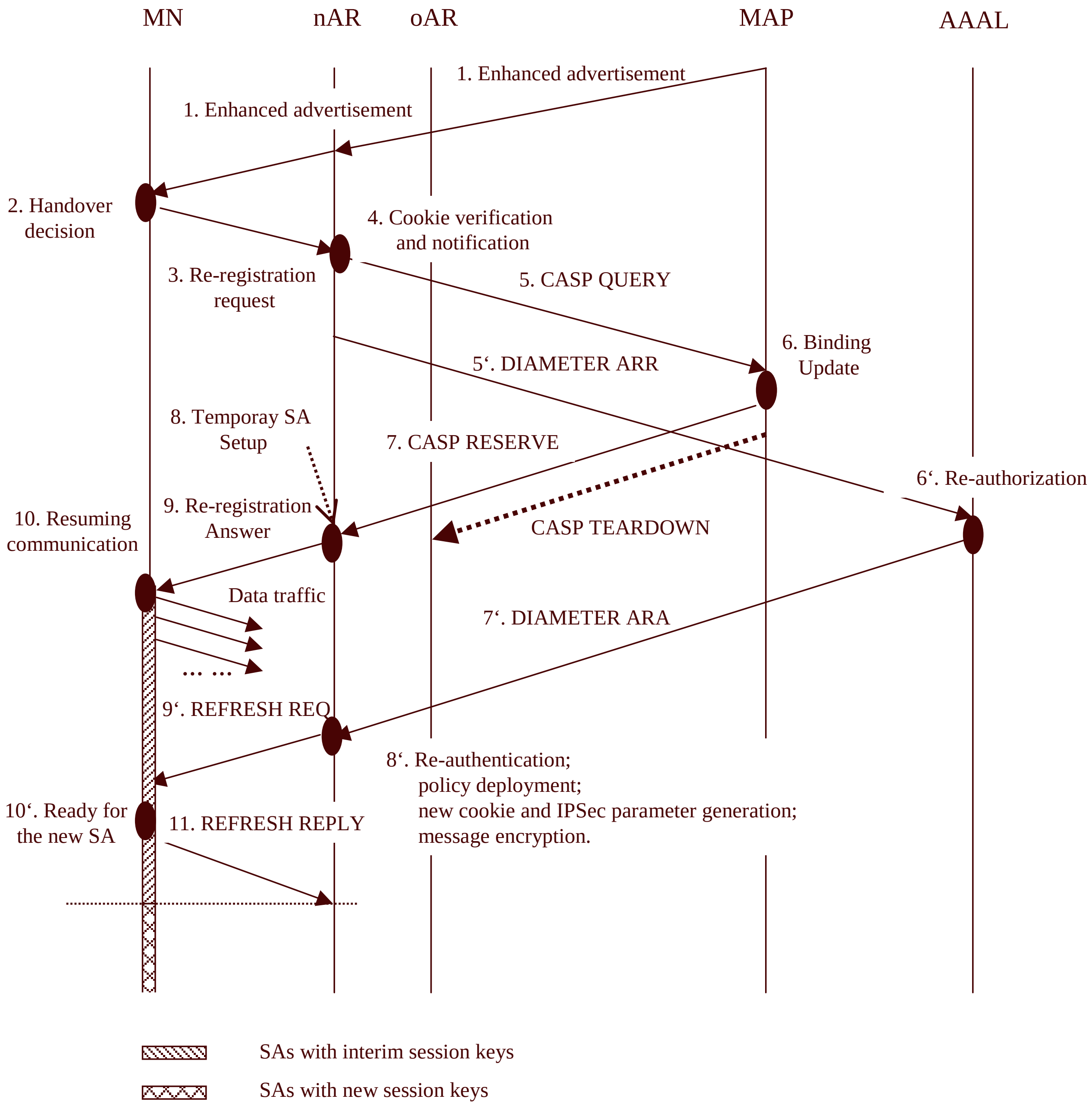

MN
nAR
oAR
MAP
AAAL
1. Enhanced advertisement
1. Enhanced advertisement
2. Handover decision
4. Cookie verification and notification
3. Re-registration request
5. CASP QUERY
6. Binding Update
5‘. DIAMETER ARR
8. Temporay SA Setup
7. CASP RESERVE
6‘. Re-authorization
10. Resuming communication
9. Re-registration Answer
CASP TEARDOWN
Data traffic
7‘. DIAMETER ARA
… …
9‘. REFRESH REQ
8‘. Re-authentication; policy deployment; new cookie and IPSec parameter generation; message encryption.
10‘. Ready for the new SA
11. REFRESH REPLY
▨ SAs with interim session keys
▨ SAs with new session keys

Figure 5.31: Message Exchanges in Re-registration When Re-authorization is Slow

Figure 5.31 shows the protocol operations in the intra-domain handover cases when the re-authorization process is slower.

When a mobile node receives multiple enhanced advertisements, it makes a handover decision based on both the content of received advertisements and its handover mode; or when the current link breaks, MN has to move to another AR after receiving an advertisement from it.

When the mobile node decides to move to a new AR (nAR), it sends a a re-registration request message to it, including a cookie, information for setting up a temporary security association, a range of a QoS parameter (e.g. bandwidth), binding update request, and information for re-authorization.

When receiving the CASP QUERY message, the access router, which is a CASP node, first verifies the cookie. The operations for the cookie verification has been discussed in Section 5.2.2. When the cookie verification is successful, the access router performs three actions simultaneously:

- generating a CASP QUERY message, which includes the range of the QoS parameter and the binding update request, and sending it to MAP;
- starting the re-authorization process with AAAL using the Diameter protocol.
- initiating a notification message to inform the ARs in the cookie's "area of validity"about use of the cookie in order to allow detection of a potential replay of the cookie;

When receiving the notification message of the used cookie, ARs record the used cookie information on a used cookie list. When receiving a request, AR will check whether the presented cookie is still on the list. The check is one of the actions of the cookie verification. When a used cookie's lifetime is running out, AR removes its information from the list.

In the integrated process of QoS+BU, the access router receives the CASP RESERVE message containing the BU ACK and the resource confirmation for the request, it sets the parameters for the temporary security associations in case that the re-authorization process has not been complete. Thus the mobile node starts the data traffic with the protection of the temporary security associations. After the MAP sends out the CASP RESERVE message to the nAR, it initiates a CASP TEARDOWN message to release the old path towards the old AR (oAR) - removing the state information for the session.

In the re-authorization process, MN includes the range of the QoS request in the Diameter AA-Registration Request (ARR) message. In case AAAL is not capable to perform the re-authorization, it communicates with the MN's AAAH for the purpose.

If the re-authorization succeeds, it updates its caches authorization information accordingly.

When the access router receives a positive answer from the re-authorization process, it takes the following actions:

- re-authenticating the request with the knowledge of the session key;
- enforcing the DiffServ policies if the service is enabled;
- generating a new cookie and related information for an IPSec security association;
- encrypting the message REFRESH REQ with the session key, and sending a message to the mobile node.

When receiving the REFRESH REQ message, the mobile node replies a REFRESH REPLY message. Afterwards, the user data is protected by a new IPSec security association.

The operation steps are listed in Table 5.1.

Steps	Function	Remarks
1	Enhanced advertisement	prefix information, aggregate available QoS information and the address of the next CASP node
2	Handover decision	handover decision based on the aggregate QoS information (e.g. Aggregate Available Bandwidth (AABW)) of a path and current mobility state (i.e. "lazy" and "eager"); generate CoAs; fetch the address of the next CASP node; make decision to make an intra-domain handover.
3	Re-registration request	message includes a range of a QoS parameter, cookie and parameter for a temporary IPSec SA, the BU request, and authentication and authorization objects.
4	cookie verification and notification	AR verifies the cookie and notifies the use of the cookie. After reserving its resource for the request, it initiates the BU+QoS process and the re-authorization process.
5	CASP QUERY	AR generates a CASP Query and sends it to MAP.
6	Binding update	MAP first checks whether the nodes along the path (including itself) can satisfy the QoS request. If yes, it registers MN's new LCoA.
7	CASP RESERVE	MAP sends a CASP Reserve to AR to confirm the reserved resource along the path. Meanwhile, it sends a CASP TEARDOWN to release the old path.

8	Temporary SA Setup	AR indicates the parameters for a temporary SA by using the interim session key.
9	CASP RESERVE	AR sends a CASP Reserve to MN.
10	Resuming communication	MN resumes the communication with CN. The user data is protected with the temporary IPSec SA. This SA is valid until a new session key comes into effect.
5’	ARR	In parallel with the BU+QoS process, AR performs a re-authorization process.
6’	Re-authorization	AAAL performs the re-authorization check based on its cached authorization information (i.e. authorization data or SLA). If the AAAL is not capable for the task, it communicates with the MN’s AAAH for the purpose. It updates the authorization data accordingly after receiving a acknowledgement message from AAAH.
7’	ARA	AAAL sends a AA-Registration Answer (ARA) message to AR.
8’	Actions when the re-authorization result arrives	the actions include authenticating the request with the session key; enforcing the policy to support DiffServ; generating a new cookie and parameters for a new IPSec security association; and encrypting the message with a session key.
9’	REFRESH REQ	AR sends a REFRESH REQ message to the MN.
10’	Ready for the new SA	when receiving the REFRESH REQ message, MN obtains the new cookie and the related information for a new IPSec security association. Then it is ready to use the new SA to protect the user data.
11	REFRESH REPLY	MN acknowledges to use the new session key for a new SA by sending a REFRESH REPLY message. When AR receives the message, the new session key comes into effect.

Table 5.1: The Signaling Procedure in the Intra-domain Case

5.3.2 Re-authorization Process Is Fast

Figure 5.31 shows the protocol operations in the intra-domain handover cases when the re-authorization process is faster. In such a case, there is no need to establish the temporary SA. Since the re-authorization process has no contribution to the registration latency in the whole procedure, the re-authentication, policy deployment and the new cookie generation and message encryption are supposed to be done completely when

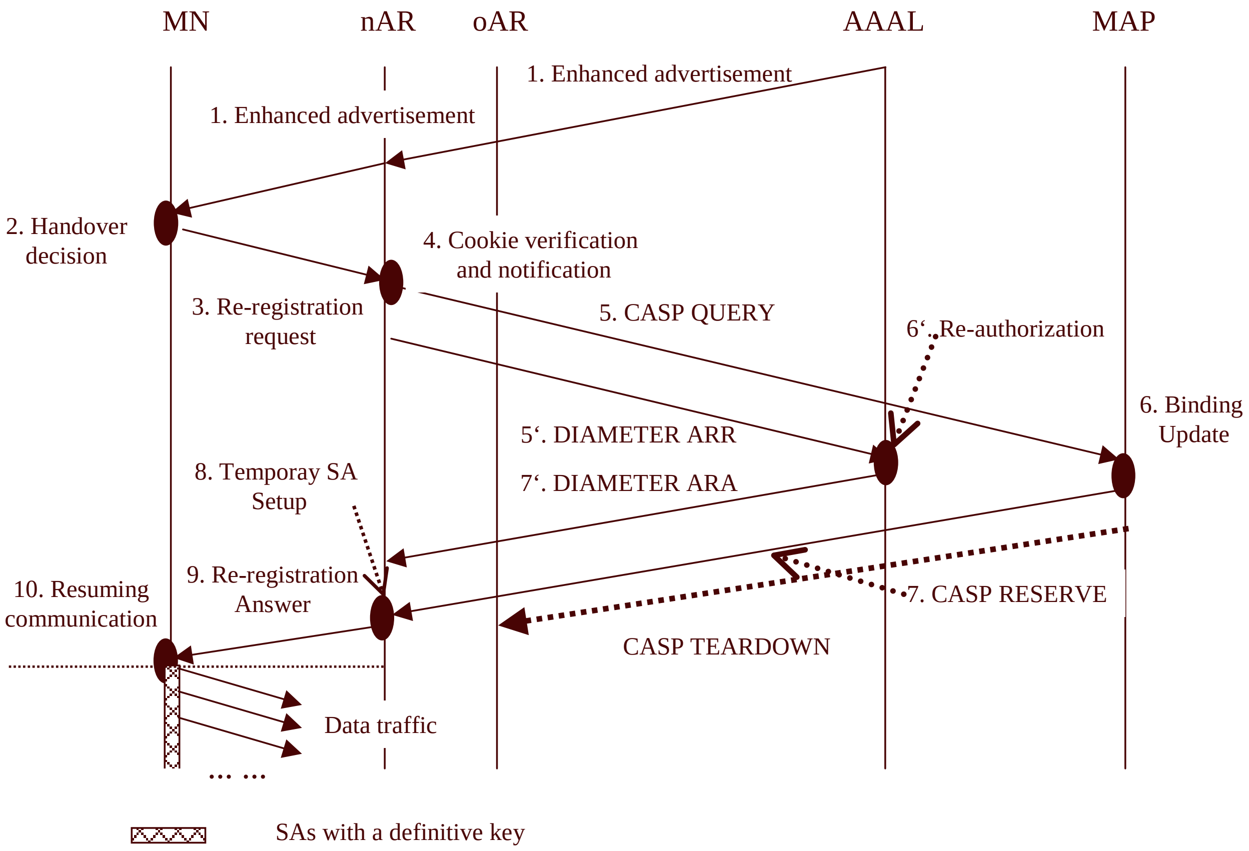

MN
nAR
oAR
AAAL
MAP
1. Enhanced advertisement
1. Enhanced advertisement
2. Handover decision
4. Cookie verification and notification
3. Re-registration request
5. CASP QUERY
6‘. Re-authorization
6. Binding Update
5‘. DIAMETER ARR
7‘. DIAMETER ARA
8. Temporay SA Setup
9. Re-registration Answer
10. Resuming communication
7. CASP RESERVE
CASP TEARDOWN
Data traffic
... ...
SAs with a definitive key

Figure 5.32: Message Exchanges in Re-authorization When Re-authorization is Fast

nAR receives the acknowledgement message from the BU+QoS process. nAR simply inserts the new cookie and parameters for a new IPSec security association in the re-registration answer message to MN. When MN receives the message, they start to use the definitive SA to protect the user data directly.

5.3.3 Summary

In the re-registration procedure, an authentication should be performed first when a request is received by an AR; There is a tradeoff between minimizing the latency in the re-registration procedure and performing a re-authorization check. Since the identity is known in the access network after the authentication check, the threats of Denial of Service (DoS) attacks and the authorization misuse are reduced, the re-authorization operations are designed to take place in parallel of other operations (i.e. binding update and resource reservation).

The binding update operation at MAP which is the only mobility management process can be integrated with the resource reservation process along the new path; The policy deployment to support DiffServ is enabled by AAAL.

Protocol Operations	MN	AR	IR	MAP	AAAL
Enhanced Advertisement	X	X	X	X	-
First Authentication - Cookie Verification	X	X	-	X	-
Temporary IPSec SA	X	X	-	X	-
CASP Mobility Client Protocol	X	X	X	X	-
Re-authorization	X	X	-	-	X
Second Authentication	X	X	-	-	X
DiffServ Support	-	X	-	-	X
Removal of the old QoS path	-	X	X	X	-
Session Update	X	X	X	X	X
BU Update	X	-	-	X	-
SA Update	X	X	-	-	X

Table 5.2: Entity Involvement in the Protocol Operations

When the re-authorization process and the BU+QoS process happen in parallel, the former one may take longer time especially when AAAH gets involved in the re-authorization (in case that MN upgrades its QoS request in the intra-domain handover and AAAL is not capable to re-authorize the request by itself). In such a case, a temporary IPSec SA is used to protect the user data during the gap of the two processes; in case the former one takes shorter time, there is no need to use a temporary SA to protect the user data.

In the following section, the protocol specications will be described for the re-registration procedure.

5.4 Protocol Specification

In this section, a high level protocol specification is described by using Specification and Description Language (SDL) [39], according to different triggering events at the involved entities.

The involvement of each entity in the protocol operations is shown in Table 5.2.

5.4.1 Mobile Node

The triggering events include receiving enhanced advertisements (Figure 5.33), a Re-registration Answer message or a Refresh Request message (Figure 5.34), and “TIME-OUT” events and others (Figure 5.35).

Reception of Enhanced Advertisements

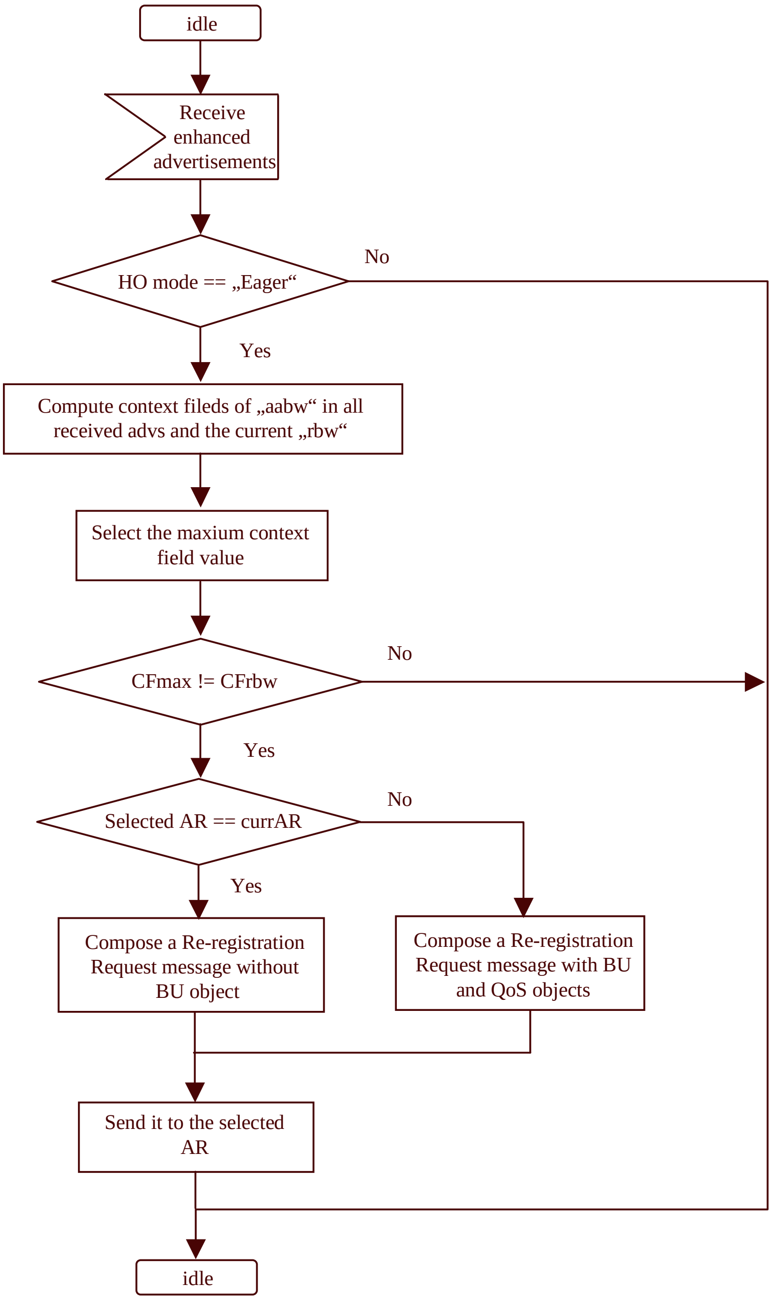

idle
Receive enhanced advertisements
HO mode == „Eager“
No
Yes
Compute context fileds of „aabw“ in all received advs and the current „rbw“
Select the maxium context field value
CFmax != CFrbw
No
Yes
Selected AR == currAR
No
Yes
Compose a Re-registration Request message without BU object
Compose a Re-registration Request message with BU and QoS objects
Send it to the selected AR
idle

Figure 5.33: MN’s Operations When Receiving Enhanced Advertisements

As shown in Figure 5.33, when receiving enhanced advertisements, if its handover mode is “eager”, it computes context field values for the Aggregate Available Bandwidth (AABW) values in all received enhanced advertisements and the current Reserved Bandwidth (RBW). If the maximum value is from RBW, MN will not take any action; otherwise, if the maximum value is from a AR which is not the current associated AR, it will make a handover composing a Re-registration message as shown in Figure 5.23

and sending it to the selected AR; if the maximum value is from the current associated AR, it will make a session update operation composing a re-registration message which includes no BU objects and sending it to the current AR.

Reception of Re-registration Answer or Refresh Request

As shown in Figure 5.34, when receiving a Re-registration Answer message or a Refresh Request message as in Figure 5.19, MN first checks the nonce. Then it check whether a cookie object is included: if yes, the received message refers to Step 6 Figure 5.20, and MN checks the signature with the session key; otherwise, the message refers to Step 5 Figure 5.19, and MN checks the signature with the temporary key.

When the received message contains a "Failure" flag, it drops the message and marks the selected AR as a "bad" one and it may try will another AR. If the re-registration request succeeds, it determines its handover mode based on whether the Reserved Bandwidth (RBW) is smaller than its Desired Bandwidth (DBW). Then MN updates the session state accordingly including caching a new cookie and new keys if necessary.

In case as shown in Figure 5.19, MN uses a temporary IPSec SA by validating the corresponding parameters; otherwise, it uses a definitive IPSec SA. If a Refresh Reply message is required, MN sends one to the associated AR indicated in the session state.

Reception of Timeout Events and Other Events

As shown in Figure 5.35, the Timeout Events include Cookie Timeout, BU Timeout, Session Timeout, SA Timeout and Session Key Timeout.

5.4.2 Access Router

The triggering events include receiving enhanced advertisements from its upper router (Figure 5.36), a Re-registration Request message (Figure 5.37), a Re-authorization Answer Message (Figure 5.40), a CASP RESERVE message (Figure 5.39), "TIMEOUT" events and others (Figure 5.41).

Reception of Enhanced Advertisements

As shown in Figure 5.36, when receiving enhanced advertisements from its upper router, AR caches a MAP option and the address of its "Upper Next CASP Node" obtained from the "next CASP node" field in an advertisement. If its own Available Bandwidth (PBW) is smaller than the Aggregate Available Bandwidth (AABW) in the advertisement, it uses its available bandwidth as the AABW in its own advertisements. Also in its own advertisements, in addition to the standard information as shown in [64], AR includes

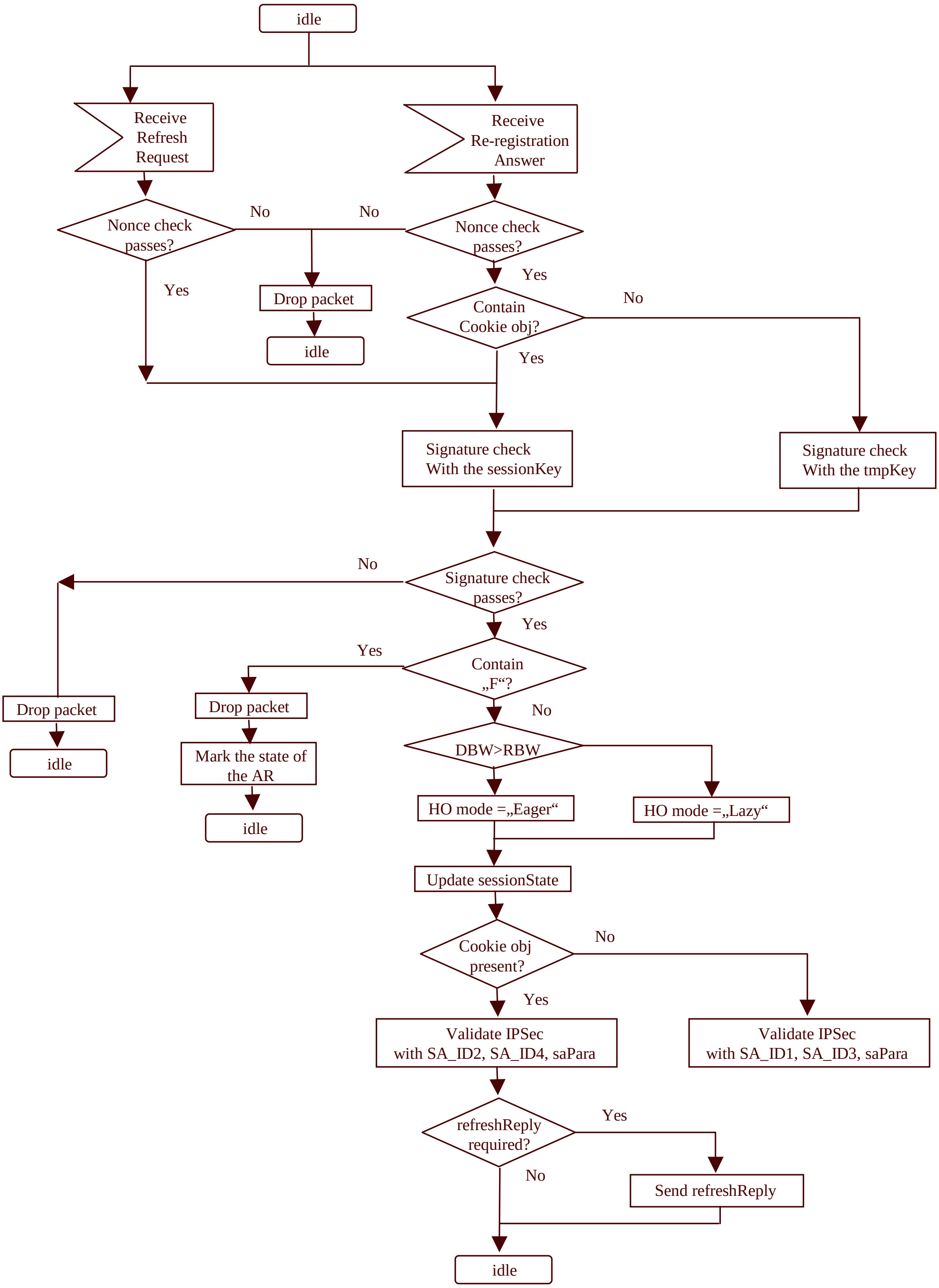


Figure 5.34: MN’s Operations When Receiving Re-reg. Ans or Refresh Request

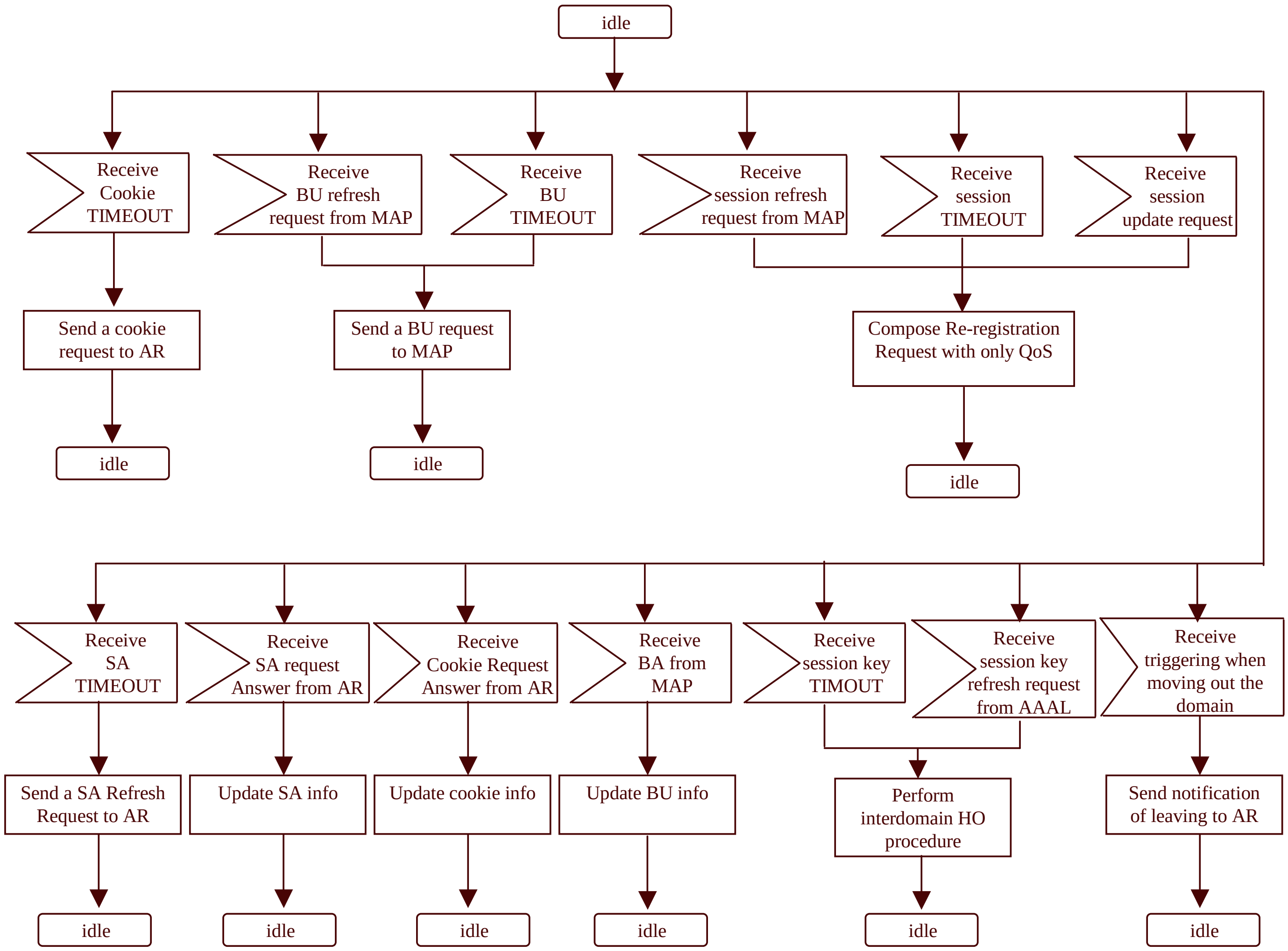

idle
Receive Cookie TIMEOUT
Receive BU refresh request from MAP
Receive BU TIMEOUT
Receive session refresh request from MAP
Receive session TIMEOUT
Receive session update request
Send a cookie request to AR
Send a BU request to MAP
Compose Re-registration Request with only QoS
idle
idle
idle
Receive SA TIMEOUT
Receive SA request Answer from AR
Receive Cookie Request Answer from AR
Receive BA from MAP
Receive session key TIMOUT
Receive session key refresh request from AAAL
Receive triggering when moving out the domain
Send a SA Refresh Request to AR
Update SA info
Update cookie info
Update BU info
Perform interdomain HO procedure
Send notification of leaving to AR
idle
idle
idle
idle
idle
idle

Figure 5.35: MN's Operations When Timeout Events Triggers and Others

the MAP options and its own address in the "next CASP node" field since it is assumed to be a CASP node in the thesis. AR broadcasts its advertisements at a pre-set rate.

Reception of Re-registration Requests

As shown in Figure 5.37, when receiving an Re-registration Request message, AR first performs a nonce check and a cookie verification. The cookie verification operations are shown in Figure 5.38. If the checks are successful, AR sends a notification of the use of the cookie to all ARs which are in the cookie's "area of validity". Also it sends a Re-authorization Request to AAAL.

If the message is a "Session Update" request, AR first creates a "Temporary Session State" relating to the current session state. The current session state is not updated by the temporary session state unless the reservation, re-authorization and re-authentication for the new request are successful.

After creating a "Temporary Session State", AR checks whether it can satisfy the

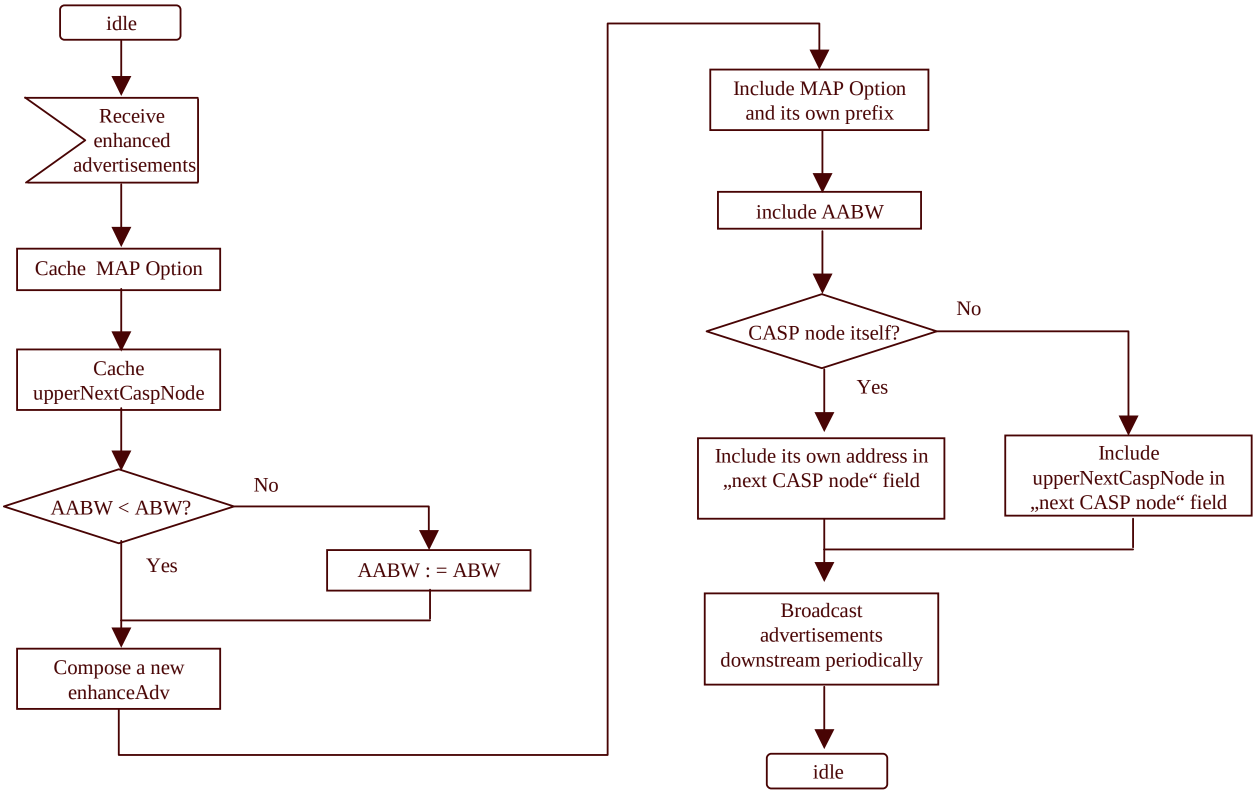

idle
Receive enhanced advertisements
Cache MAP Option
Cache upperNextCaspNode
AABW < ABW?
No
Yes
AABW : = ABW
Compose a new enhanceAdv
Include MAP Option and its own prefix
include AABW
CASP node itself?
No
Yes
Include its own address in „next CASP node“ field
Include upperNextCaspNode in „next CASP node“ field
Broadcast advertisements downstream periodically
idle

Figure 5.36: AR's Operations When Receiving Enhanced Advertisements

QoS range. If no, it will set a "Failure" flag in the temporary session state when the re-authorization has not finished; it will remove the temporary session state and send a Re-registration Answer Message with a "Failure" flag to MN, including a new cookie which is encrypted with the session key; if yes, AR adjusts the Desired Bandwidth (DBW) if its PBW is smaller than the DBW in the QoS range, and it sends a CASP QUERY message to its "Upper Next CASP Node" including the new QoS range.

If the message is "Handover" request, AR creates a new session state (called "Current Session State"). Then the other operations are the same as in the case of the "Session Update" request.

Reception of CASP RESERVE

As shown in Figure 5.39, when receiving a "CASP RESERVE" message, AR will drop it if the session does not exist any longer.

If the message corresponds to a "session update" request, AR first checks whether a "Failure" flag is set in the message. If the "Failure" flag is set, AR sets "qosbuFlag" to "Failure". In case that the Re-authorization process has finished, AR removes the "Temporary Session State" and includes a new cookie encrypted with the session key in a Re-registration Answer message which has a "Failure" flag.

If the "CASP RESERVE" is a successful one, AR updates the "RBW" in the "Tem-

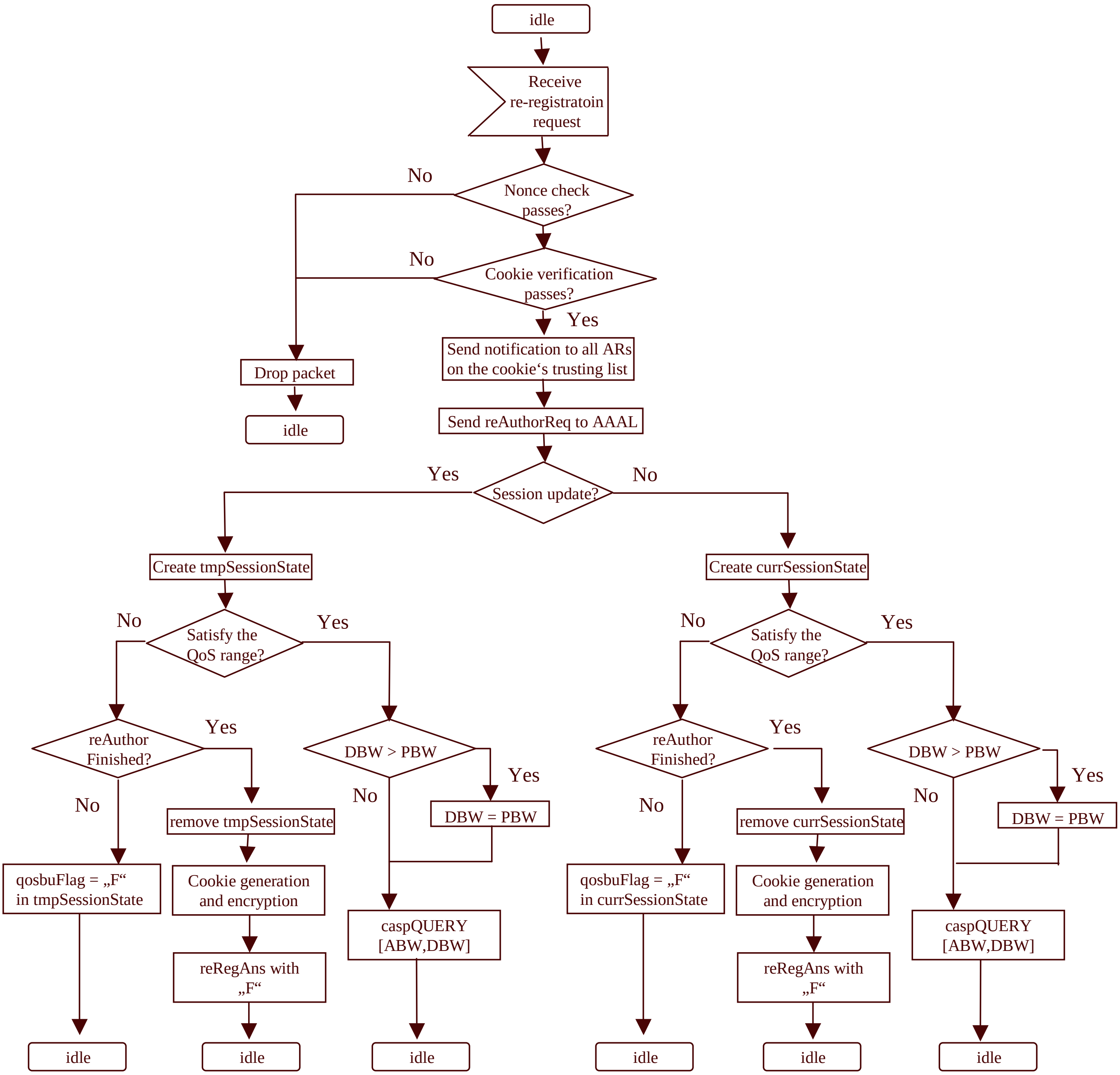


Figure 5.37: AR's Operations When Receiving an Re-registration Request Message

porary Session State" as indicated in the "CASP RESERVE" message.

In case that the Re-authorization process has not finished, AR reserves resources as the "RBW" and sends a "Re-registration Answer" message as Figure 5.24 in Step 5 in Figure 5.19. In order to resume the "Current Session State" if the authorization fails, the "Current Session State" is saved in "Old Session State". Then the "Current Session State" is updated by "Temporary Session State" and the "Temporary Session State" is removed.

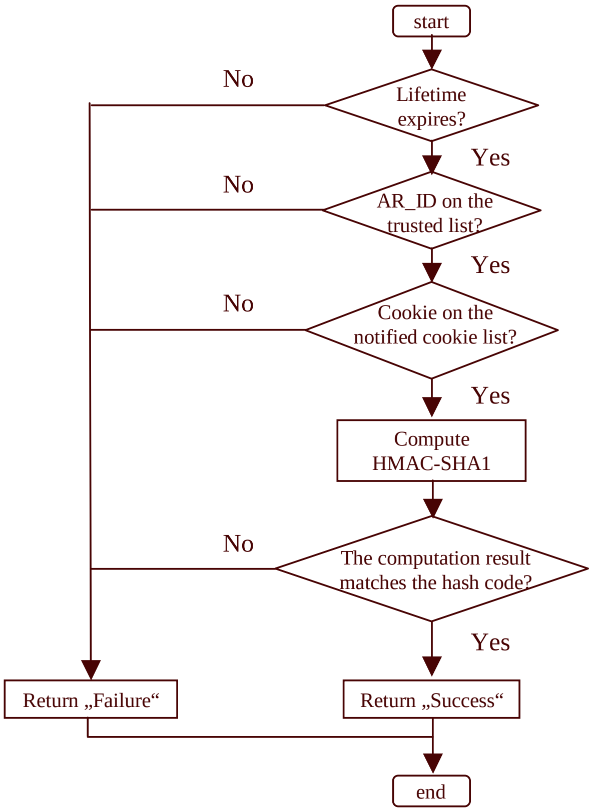

start
Lifetime expires?
No
Yes
AR_ID on the trusted list?
No
Yes
Cookie on the notified cookie list?
No
Yes
Compute HMAC-SHA1
The computation result matches the hash code?
No
Yes
Return „Failure“
Return „Success“
end

Figure 5.38: AR's Operations for Cookie Verification

In case that the Re-authorization process has finished, when the re-authorization is successful, AR reserves resources as the "RBW" and sends a "Re-registration Answer" message as shown in Step 6 in Figure 5.20. AR then updates the "Current Session State" with the "Temporary Session State". When the re-authorization is not successful, AR sends a new "CASP QUERY" message containing the QoS range in its "Current Session State" to its "Upper Next CASP Node" in order to resume the current reservation on the path. Then it removes the "Temporary Session State" and sends a negative Re-registration Answer to MN along with an encrypted cookie.

In the "Handover" case, if the "CASP RESERVE" fails, AR sets the "qosbuFlag" in the "Current Session State" as "Failure". If re-authorization process has finished, AR removes the session state and sends a negative Re-registration Answer to MN along with an encrypted cookie.

If the "CASP RESERVE" succeeds, AR first updates the "RBW" in the "Current Session State". In case that the "re-authorization" has not finished, it reserves resources as "RBW", and sends a "Re-registration Ans" message as Figure 5.24 in Step 5 in Figure 5.19; in case that the "re-authorization" has finished with a "Success" flag, AR reserves resources as the "RBW", deploys the "DiffServ Policy" and sends a "Re-registration Answer" message as shown in Step 6 in Figure 5.20; in case that the "re-authorization" has

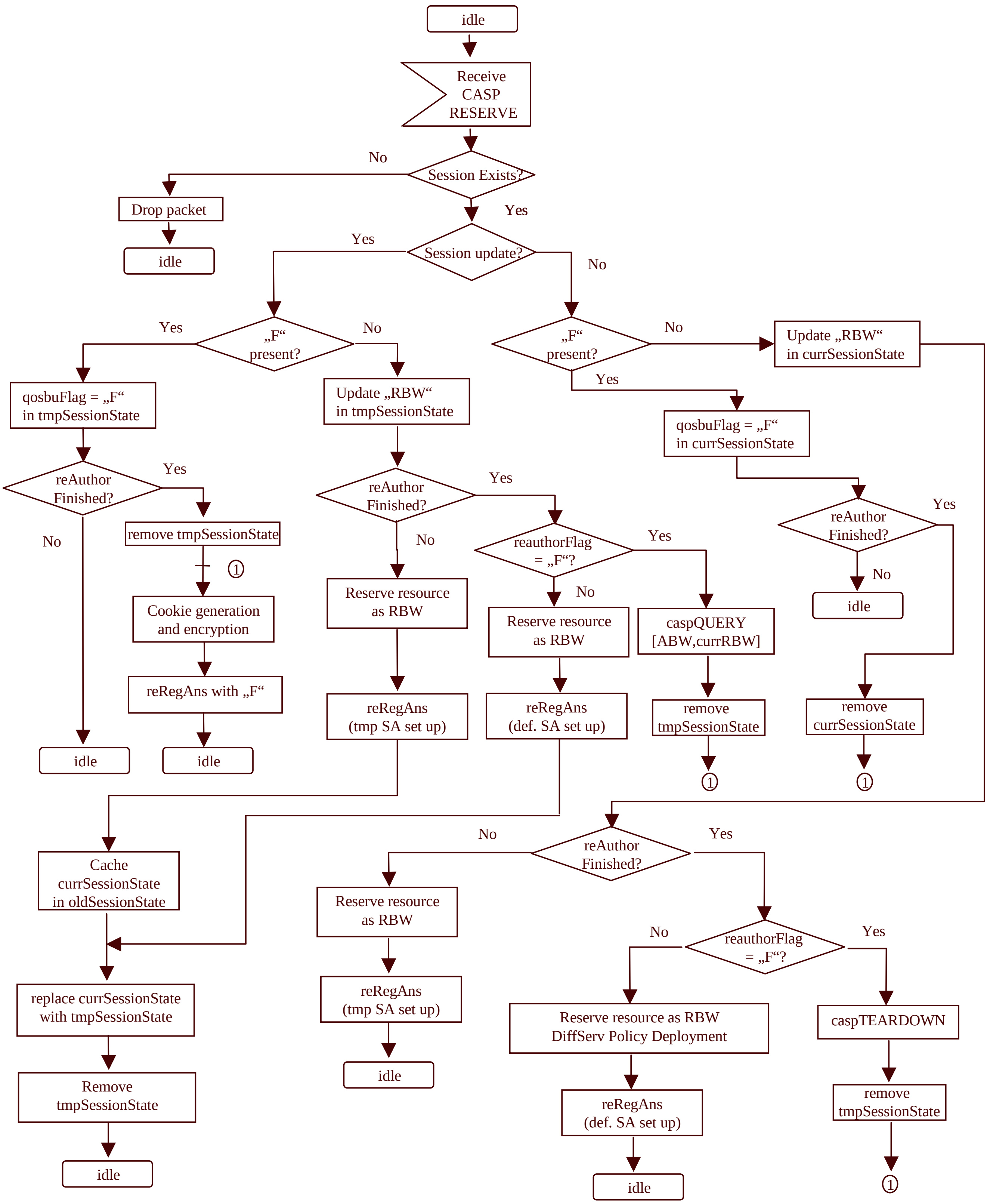


Figure 5.39: AR’s Operations When Receiving a CASP RESERVE Message

finished with a “Failure” flag, ARs sends a “CASP TEARDOWN” message to remove the reservation on the path. Meanwhile, it sends a negative “Re-registration Answer” with a encrypted cookie.

Reception of Re-authorization Answer

As shown in Figure 5.40, when receiving a “CASP RESERVE” message, AR will drop it if the session does not exist.

In case that the “re-authorization” fails, if the QoS+BU process has not finished (i.e. the corresponding CASP RESERVE has not arrived), AR sets the “authorFlag” to “Failure” in the “Current Session State” or “Temporary Session State” in case of “Session Update” or “Handover” respectively; if the QoS+BU process has finished unsuccessfully, AR removes the temporary session state or current session state in case of “Session Update” or “Handover” case respectively. Then AR sends a “Re-registration Answer” message with a “Failure” to MN along with a new cookie; if the QoS+BU process has finished successfully, in case of “Session update”, AR sends a “CASP QUERY” message containing the QoS range in the “oldSessionState” to resume the old reservation. Then AR informs MN about the failure of the “Session Update” request while in case of “Handover”, AR sends a “CASP TEARDOWN” message to remove the reservation along the new path.

In case that the “re-authorization” succeeds, AR first performs a re-authentication check with the session key. If the re-authentication fails, ARs sends a “CASP TEARDOWN” message along the path, and then removes all the relevant session states. Otherwise, it checks whether the QoS+BU process has finished. If no, AR sets the authorFlag to “Success” and cache “DiffServ Policy” in the current session state in case of “Handover” or in the temporary session state in case of “Session Update”.

If the re-authorization has finished unsuccessfully, AR removes the temporary session state in case of “Session Update” or the current session state in case of “Handover”. And then it acknowledges MN about the failure.

If the re-authorization has finished successfully, in case of “Session Update”, if Subscribed Bandwidth (SBW) is not presented in the message which means the DBW is authorized, AR removes the old session state in case of, deploys the “DiffServ Policy” and composes a “Refresh Request” message as shown in Figure 5.25 in Step 7 of Figure 5.19 except the BA object; if Subscribed Bandwidth (SBW) is presented in the message which means the DBW is not authorized, AR may send a “CASP QUERY” message on the path to reduce the reserved resource to SBW if resource has been reserved more than what the MN is entitled. The “CASP QUERY” message does not require a “Re-registration Answer” message being sent to MN.

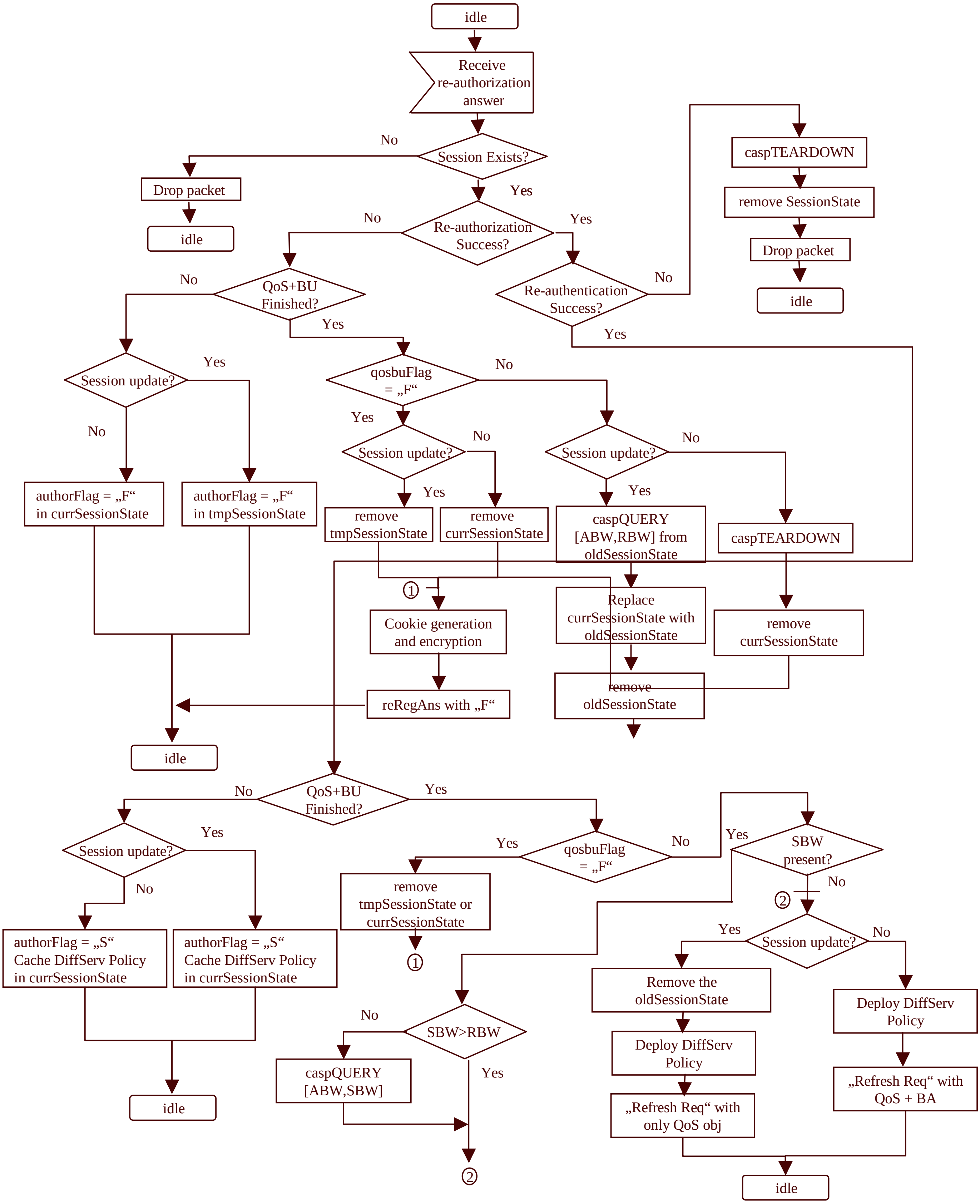

idle
Receive re-authorization answer
Session Exists?
No
Yes
Drop packet
idle
Re-authorization Success?
No
Yes
caspTEARDOWN
remove SessionState
Drop packet
idle
QoS+BU Finished?
No
Yes
Re-authentication Success?
No
Yes
Session update?
Yes
No
qosbuFlag = „F“
No
Yes
Session update?
No
Yes
Session update?
No
Yes
authorFlag = „F“ in currSessionState
authorFlag = „F“ in tmpSessionState
remove tmpSessionState
remove currSessionState
caspQUERY [ABW,RBW] from oldSessionState
caspTEARDOWN
1
Cookie generation and encryption
Replace currSessionState with oldSessionState
remove currSessionState
reRegAns with „F“
remove oldSessionState
idle
QoS+BU Finished?
No
Yes
Session update?
Yes
No
qosbuFlag = „F“
Yes
No
SBW present?
Yes
No
2
authorFlag = „S“ Cache DiffServ Policy in currSessionState
authorFlag = „S“ Cache DiffServ Policy in currSessionState
remove tmpSessionState or currSessionState
1
Session update?
Yes
No
Remove the oldSessionState
Deploy DiffServ Policy
SBW>RBW
No
Yes
caspQUERY [ABW,SBW]
Deploy DiffServ Policy
„Refresh Req“ with only QoS obj
„Refresh Req“ with QoS + BA
idle
2
idle

Figure 5.40: AR’s Operations When Receiving a Re-authorization Answer Message

Reception of Timeout Events and Other Events

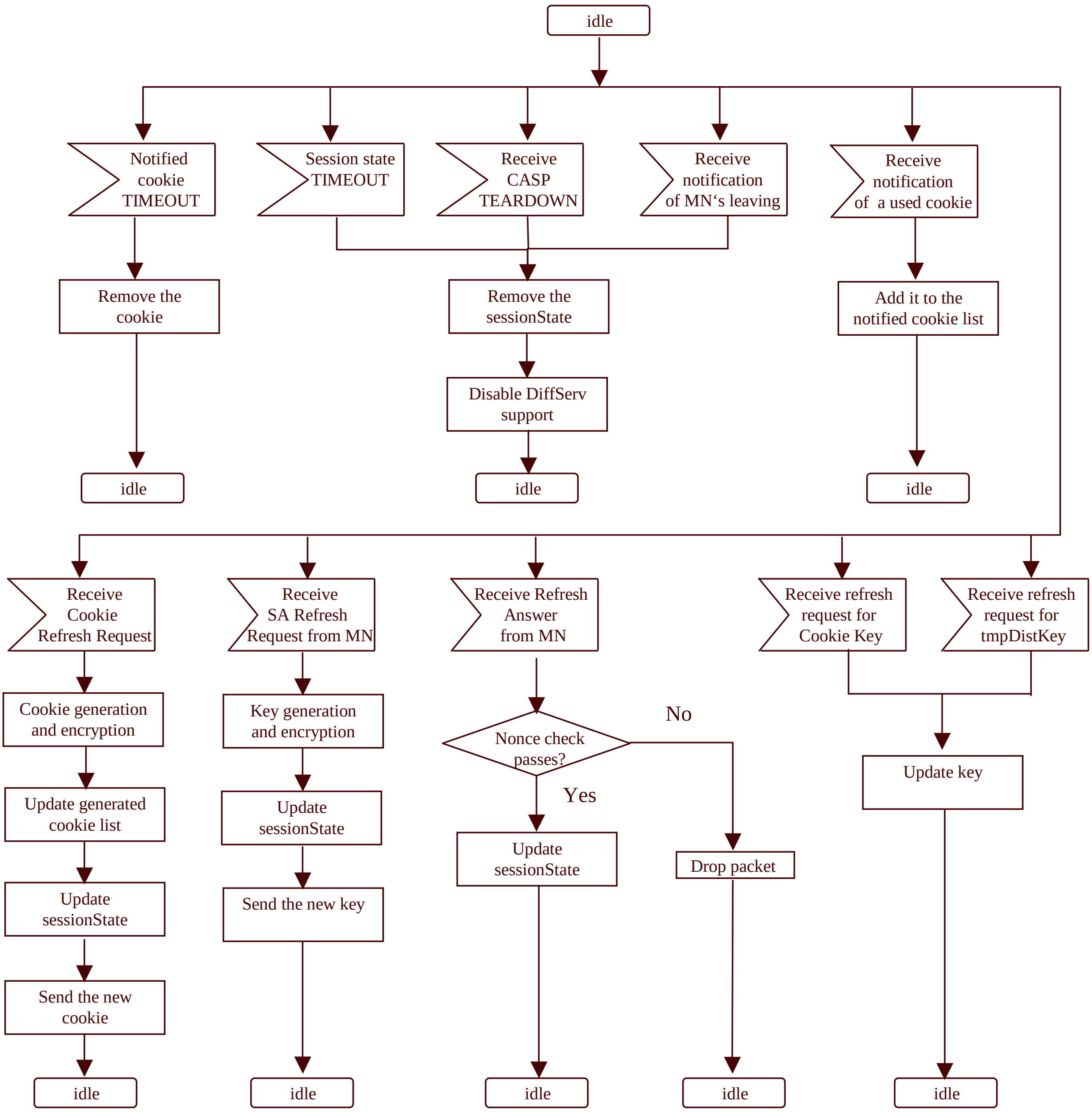

idle
Notified cookie TIMEOUT
Session state TIMEOUT
Receive CASP TEARDOWN
Receive notification of MN's leaving
Receive notification of a used cookie
Remove the cookie
Remove the sessionState
Add it to the notified cookie list
Disable DiffServ support
idle
idle
idle
Receive Cookie Refresh Request
Receive SA Refresh Request from MN
Receive Refresh Answer from MN
Receive refresh request for Cookie Key
Receive refresh request for tmpDistKey
Cookie generation and encryption
Key generation and encryption
Nonce check passes?
No
Yes
Update key
Update generated cookie list
Update sessionState
Update sessionState
Drop packet
Update sessionState
Send the new key
Send the new cookie
idle
idle
idle
idle
idle

Figure 5.41: AR's Operations When Timeout Events Triggers and Others

As shown in Figure 5.41, the Timeout Events include notified cookie Timeout, Session Timeout. ARs need to take actions when the following events trigger: receiving a "CASP TEARDOWN", "Notification of Leaving of MN", "Notification of a Used Cookie", "Cookie Refresh Request from MN", "SA Refresh Request from MN", "Refresh Answer", "Cookie Key Refresh from MAP" and "Temporary Distributed Key Refresh from MAP".

5.4.3 Intermediate Router

The triggering events include receiving enhanced advertisements from its upper router (Figure 5.36), a CASP QUERY message (Figure 5.42), a CASP RESERVE message (Figure 5.43), “TIMEOUT” events and a CASP TEARDOWN message (Figure 5.44).

Reception of Enhanced Advertisements

The operations of an Intermediate Router (IR) are the same as those of an Access Router (AR) as shown in Figure 5.36.

Reception of CASP QUERY

As shown in Figure 5.42, when receiving a “CASP QUERY” message, IR first checks the session ID of the request. If it knows the session ID, either it needs to process the message as a “Session Update” request, or it should act as the Cross-over Router (CR) (i.e. the joint point of the old path and the new path) to process the “QoS+BU” message; if it does not know the session ID, it should act as a normal intermediate router to process the “QoS+BU” message.

In the “Session Update” case, the IR checks whether it can satisfy the QoS range. If no, it composes a “CASP RESERVE” message with a “Failure” flag and sends it to the sender of the message. Thus, the old session is unchanged; if yes, it modifies the Desired Bandwidth (DBW) to the bandwidth value it can provide if its Available Bandwidth (PBW) is smaller than the DBW. Then it creates a new “Temporary Session State” filling in the relevant information. Finally it composes a new “CASP QUERY” message with the new QoS range and sends it to its “Upper Next CASP Node”.

In case that the IR acts as the Cross-over Router (CR), the IR also checks whether it can satisfy the QoS range first. If no, it does the same as in the “Session Update” case. It keeps the old path unchanged; if yes, it also creates a “Temporary Session State” with the relevant information. Then it composes a “CASP RESERVE” message with only the BU object and sends it to MAP directly. Meanwhile, it starts a timer waiting for the BA from MAP.

In case that the IR is a normal intermediate router, if it can not satisfy the QoS range, it does the same as in the “Session Update” case. It thus has no state being established for the request; otherwise, it creates a new session state after modifying the DBW if necessary. Then it composes a new “CASP QUERY” message containing both BU and QoS objects and sends it.

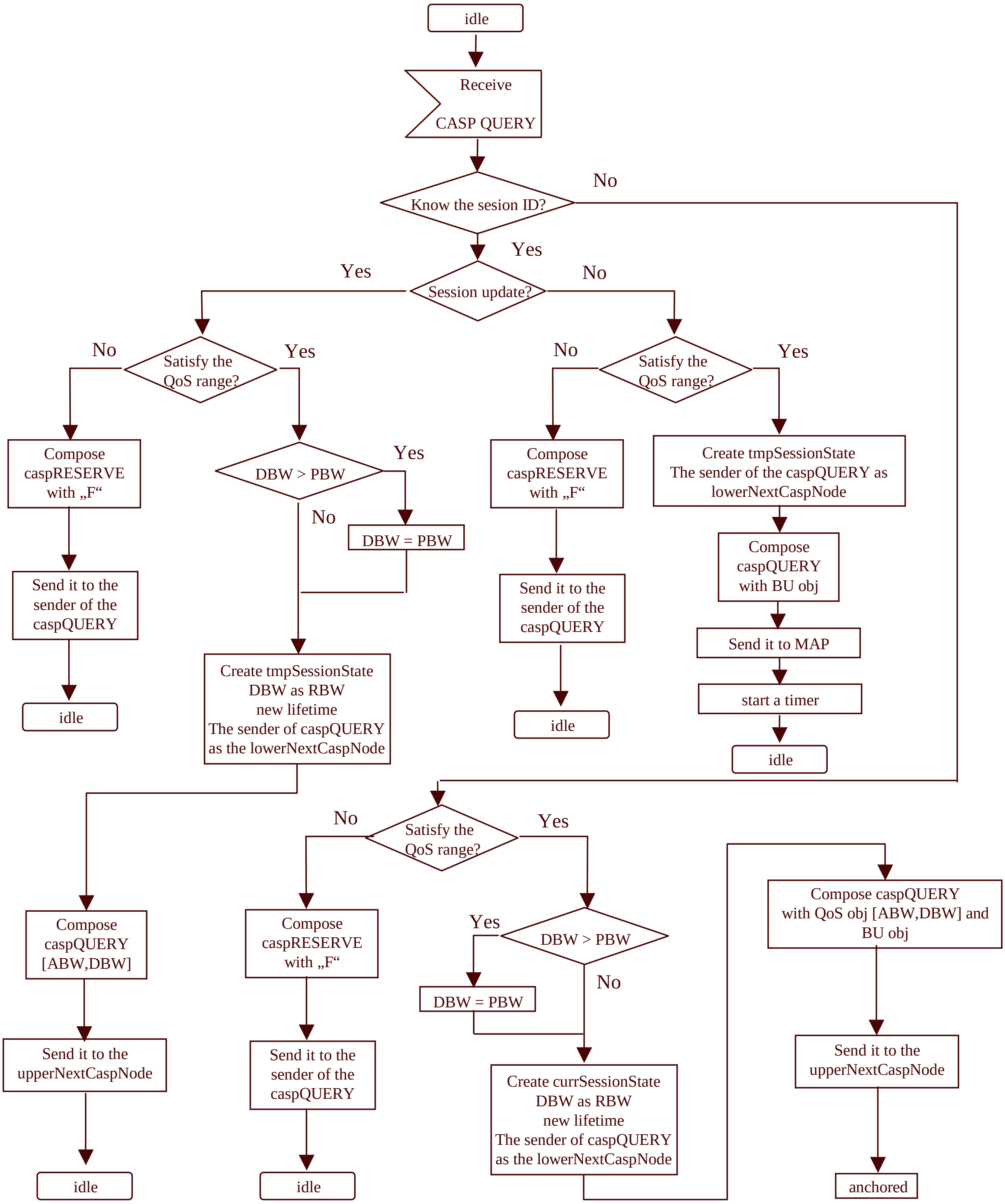

idle
Receive
CASP QUERY
Know the sesion ID?
No
Yes
Session update?
Yes
No
Satisfy the QoS range?
No
Yes
Compose caspRESERVE with „F“
Send it to the sender of the caspQUERY
idle
DBW > PBW
Yes
No
DBW = PBW
Create tmpSessionState DBW as RBW new lifetime The sender of caspQUERY as the lowerNextCaspNode
Satisfy the QoS range?
No
Yes
Compose caspRESERVE with „F“
Send it to the sender of the caspQUERY
idle
Create tmpSessionState The sender of the caspQUERY as lowerNextCaspNode
Compose caspQUERY with BU obj
Send it to MAP
start a timer
idle
Satisfy the QoS range?
No
Yes
Compose caspQUERY [ABW,DBW]
Send it to the upperNextCaspNode
idle
Compose caspRESERVE with „F“
Send it to the sender of the caspQUERY
idle
DBW > PBW
Yes
No
DBW = PBW
Create currSessionState DBW as RBW new lifetime The sender of caspQUERY as the lowerNextCaspNode
Compose caspQUERY with QoS obj [ABW,DBW] and BU obj
Send it to the upperNextCaspNode
anchored

Figure 5.42: IR’s Operations When Receiving a CASP QUERY Message

Reception of CASP RESERVE

As shown in Figure 5.43, when receiving a “CASP RESERVE” message, the IR first checks whether the session still exists. It then checks whether the “Failure” flag in the

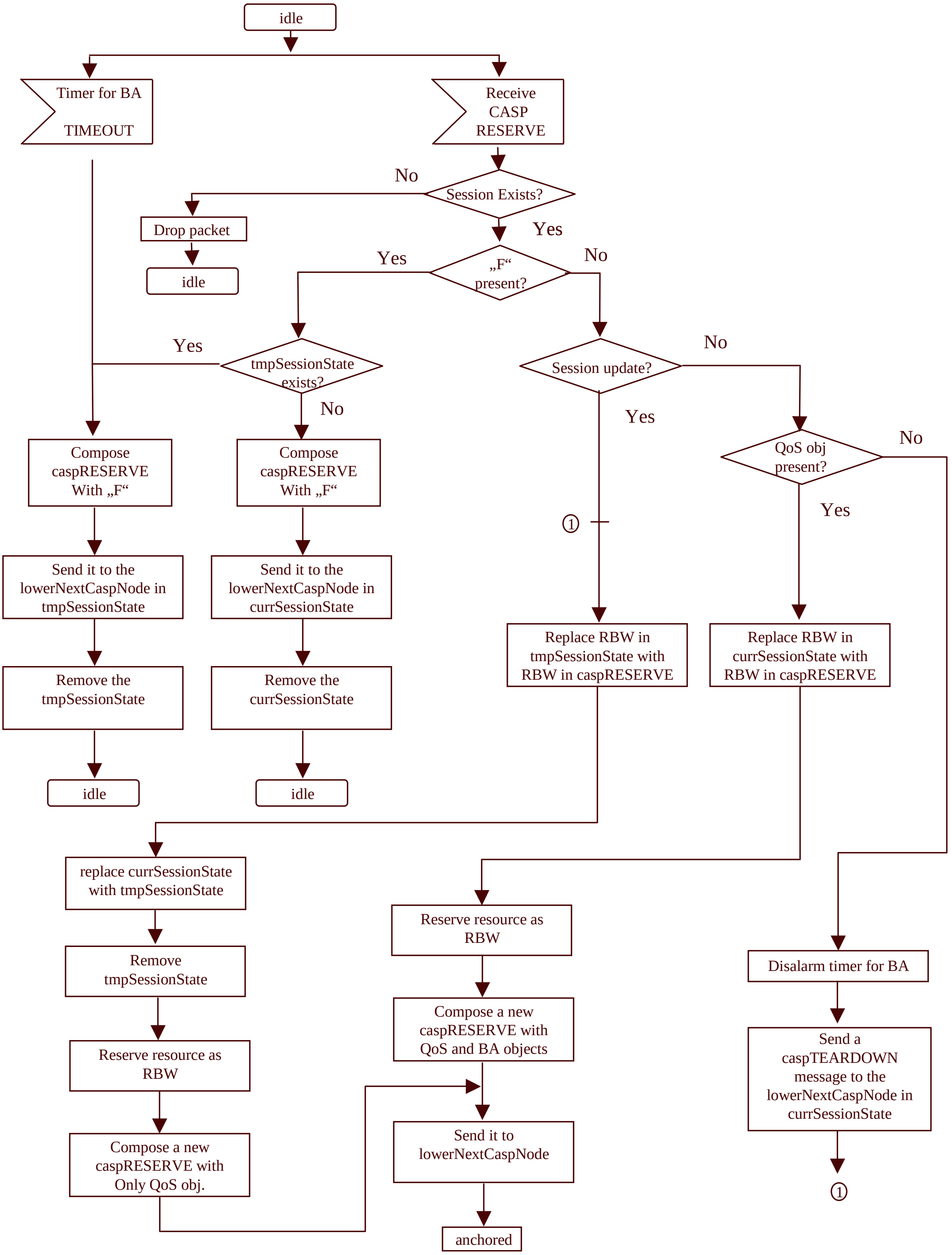


Figure 5.43: IR’s Operations When Receiving a CASP RESERVE Message

message is set.

In case that the received “CASP RESERVE” message contains a “Failure” flag, if the “Temporary Session State” exists, it sends the “Lower Next CASP Node” indicated in the “Temporary Session State” a failed CASP RESERVE message. Finally it removes the temporary session state to keep the old session (in case of “Session Update”) or the old path (in case that IR is a “CR”); if not, it sends a failed CASP RESERVE message to the “Lower Next CASP Node” indicated in the “Current Session State”, and removes the current session state.

In case that the “CASP RESERVE” message is successful, it operates differently in case of “Session Update”, “IR” and “CR”.

In case of “Session Update”, IR first replaces the RBW in the “Temporary Session State” the RBW value in the received “CASP RESERVE” message. Then it validates the temporary session state to the current one. Finally it reserves resources, composes a new “CASP RESERVE” message containing only the QoS object and sends it.

In case of “IR” (i.e. QoS object is included in the received “CASP RESERVE” message), IR also replaces the RBW in the “Temporary Session State” the RBW value in the received “CASP RESERVE” message. Then it reserves resources, composes a new “CASP RESERVE” message containing both the QoS and BA objects and sends it.

In case of “CR”, IR first disalarms the timer for the BA. Then it sends a “CASP TEARDOWN” message to the “Lower Next CASP Node” in the “Current Session State”. Afterwards, it operates as in the “Session Update” case.

Reception of Session State TIMEOUT and CASP TEARDOWN

As shown in Figure 5.44, when the session state times out, it removes the state. When receiving a “CASP TEARDOWN” message, IR informs its next CASP node to remove the session while it removes the session itself.

5.4.4 Mobility Anchor Point

The triggering events include sending enhanced advertisements (Figure 5.45), receiving a CASP QUERY message (Figure 5.46), and Timeout events and others (Figure 5.47).

Reception of Enhanced Advertisements

The operations of sending enhanced advertisements are shown in Figure 5.45.

Reception of CASP Query

Figure 5.46 shows the MAP’s operations when receiving a CASP QUERY message. In case of “Session Update”, MAP first checks whether it can satisfy the QoS range. If

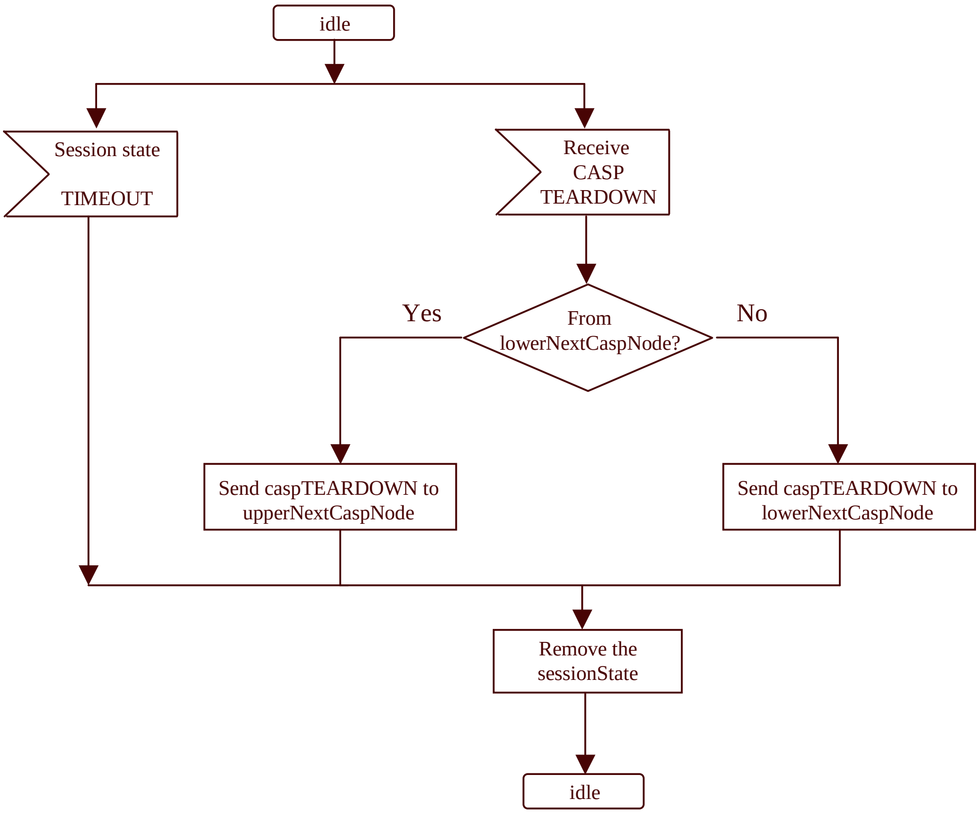


Figure 5.44: IR's Operations When Receiving CASP RESERVE and Other Events

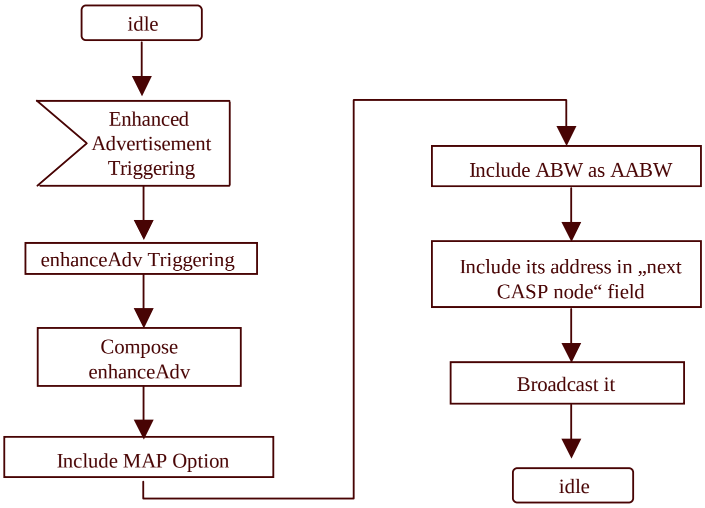


Figure 5.45: MAP's Operations of Sending Enhanced Advertisements

no, it composes a "CASP RESERVE" message with a "Failure" flag and send it to the sender of the "CASP QUERY" message; if yes, it may adjust the "DBW" to its "PBW" if its available bandwidth is smaller than the "DBW" value. It then reserves resources as "DBW" and updates the Reserved Bandwidth (RBW) in the "Current Session State". Finally it composes a "CASP RESERVE" message including the "RBW" and sends it.

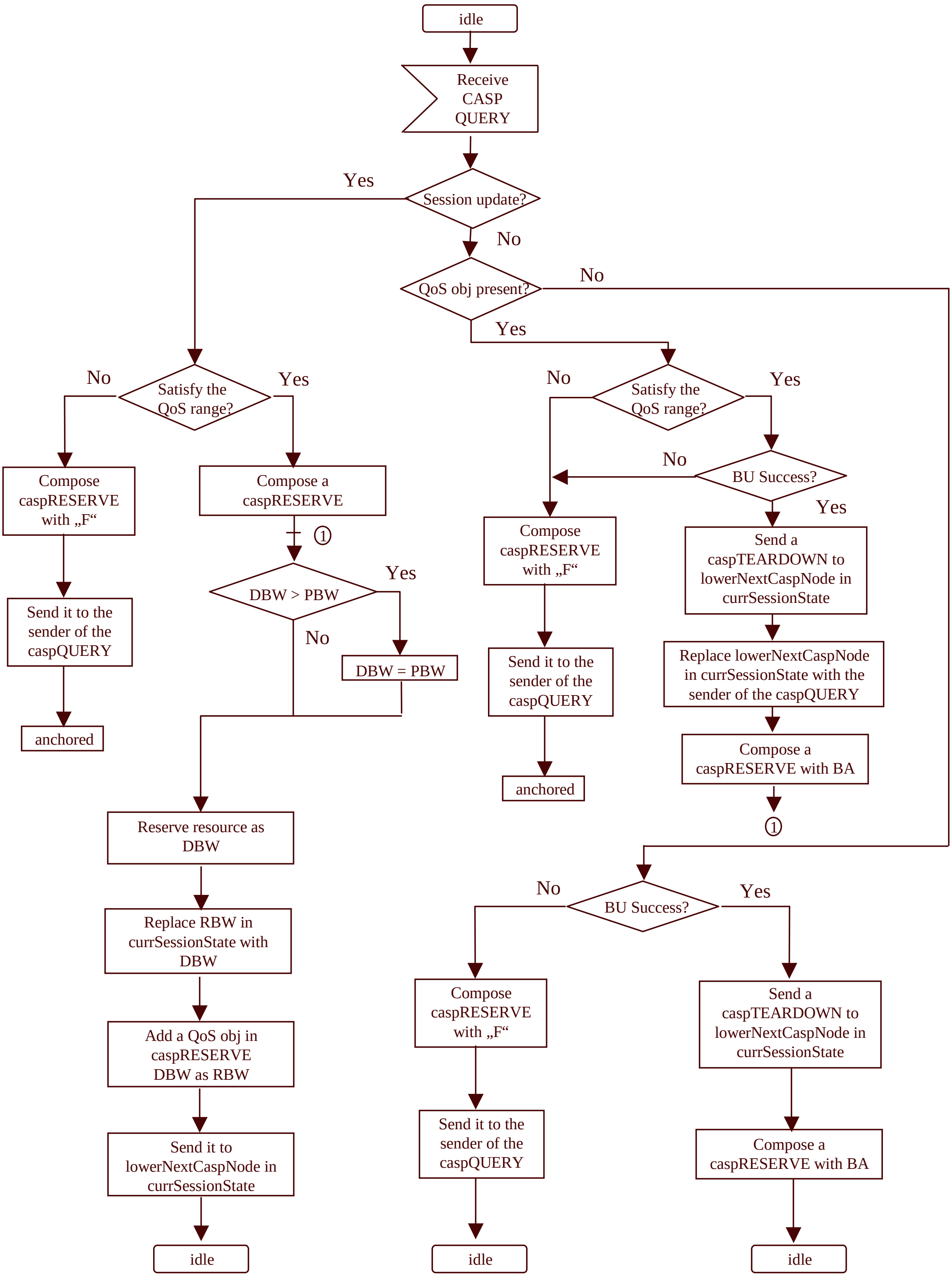

idle
Receive CASP QUERY
Session update?
Yes
No
QoS obj present?
No
Yes
No
Satisfy the QoS range?
Yes
No
Satisfy the QoS range?
Yes
Compose caspRESERVE with „F“
Compose a caspRESERVE
1
No
BU Success?
Yes
Compose caspRESERVE with „F“
Send a caspTEARDOWN to lowerNextCaspNode in currSessionState
Yes
DBW > PBW
No
DBW = PBW
Send it to the sender of the caspQUERY
Send it to the sender of the caspQUERY
Replace lowerNextCaspNode in currSessionState with the sender of the caspQUERY
anchored
Compose a caspRESERVE with BA
anchored
1
Reserve resource as DBW
No
BU Success?
Yes
Replace RBW in currSessionState with DBW
Compose caspRESERVE with „F“
Send a caspTEARDOWN to lowerNextCaspNode in currSessionState
Add a QoS obj in caspRESERVE DBW as RBW
Send it to lowerNextCaspNode in currSessionState
Send it to the sender of the caspQUERY
Compose a caspRESERVE with BA
idle
idle
idle

Figure 5.46: MAP’s Operations When Receiving a CASP QUERY Message

In case of “Handover”, if a “QoS Request Object” is presented, MAP acts as a “CR”. If MAP can not satisfy the QoS range, it sends a “CASP RESERVE” message with a “Failure” flag without changing the current session state; otherwise, it performs a binding update operation. If the BU operation is successful, it sends a “CASP TEARDOWN” to the “Lower Next CASP Node” in the “Current Session State” to release the old path. It then changes the the “Lower Next CASP Node” to the sender of the “CASP QUERY” message. Finally it composes a “CASP RESERVE” message with a “BA Object” and performs the same operations as in the “Session Update” case. If the received “CASP RESERVE” message does not contain a “QoS Request Object”, only a “BU Object”, it first performs a binding update operation. If the BU operation is successful, it sends a “CASP TEARDOWN” to the “Lower Next CASP Node” in the “Current Session State” to release the old path. Then it composes a “CASP RESERVE” with only a “BA Object” and sends it to the sender of the “CASP QUERY” message.

Reception of Timeout Events and Other Events

As shown in Figure 5.47, the timeout events include BU Timeout, Session State Timeout, Cookie Key Timeout, Temporary Distribution Key Timeout.

When its timers alarm due to expiration of the BU or the Session State, MAP sends a notification to MN.

Other triggering events include receiving a “CASP TEARDOWN” message, a “BU” message from MN, or a “TEARDOWN” request from AAAL.

5.4.5 The Local AAA Server

The triggering events including receiving a Re-authorization Request message from a AR, a Re-authorization Answer message from a MN’s AAAH are shown in Figure 5.48. The Timeout Events and others are also shown in the figure.

Reception of Re-authorization Request

When receiving a “Re-authorization Request” message from a AR, AAAL takes out the corresponding authorization data and compares the “DBW” with the Cached Bandwidth Value at AAAL (CBW) value. If the CBW is greater, AAAL sends a “Re-authorization Answer” message including a “Success” flag as the re-authorization result, the session key and MN’s DiffServ policy; otherwise, AAAL sends a “Re-authorization Request” to the MN’s AAAH and starts a timer waiting for the corresponding answer.

Alternatively, when the AAAH gets involved in the re-authorization, the AAAL may first send the session key to the AR so that the AR can perform the second-step authentication without waiting for the result of the remote re-authorization process which may

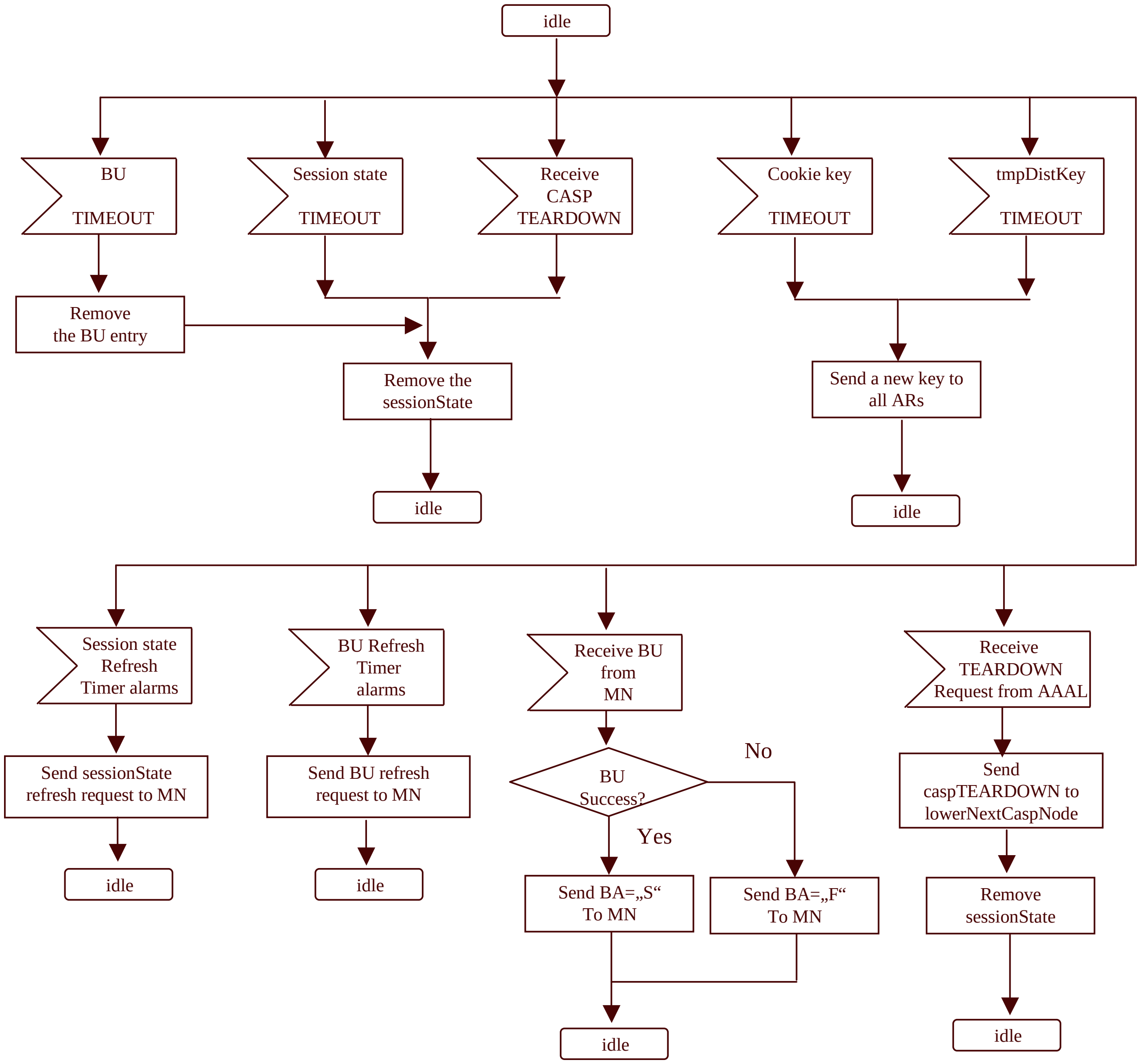


Figure 5.47: MAP's Operations When Timeout Events Trigger and Others

take a long time. Thus, AR is able to remove a false reservation along the path for a bogus request (which might gain access with a "replay" cookie) once the second-step authentication fails. More discussion is available in the following chapter - "Security Analysis of DoS".

Reception of Re-authorization Answer

When receiving a re-authorization answer from the AAAH, AAAL disalarms the timer first. If the remote re-authorization is successful and the Subscribed Bandwidth (SBW) is presented, AAAL first updates the authorization data with the SBW value, and then includes SBW in the re-authorization answer message; if the remote re-authorization is

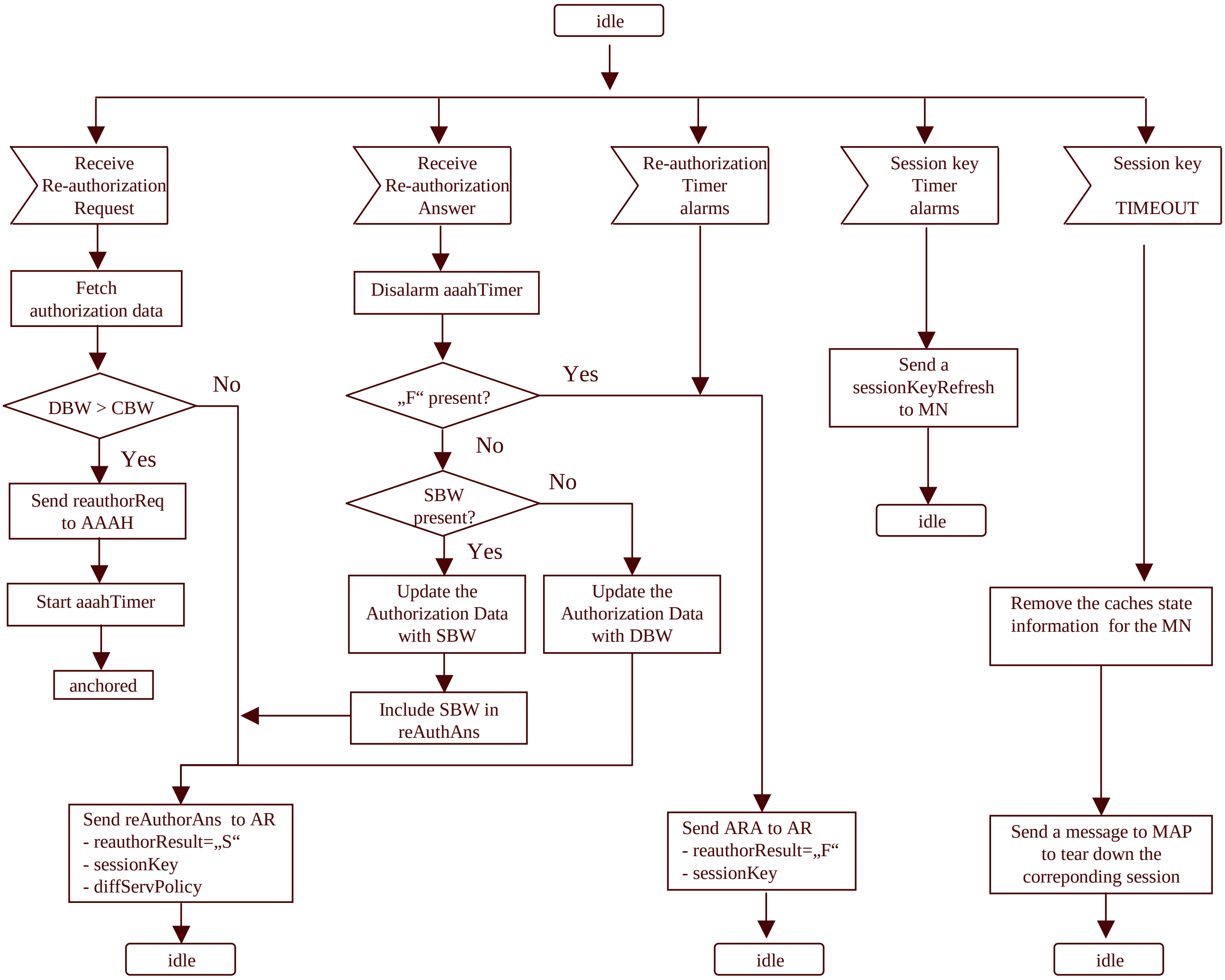


Figure 5.48: AAAL's Operations

successful and the Subscribed Bandwidth (SBW) is not presented, AAAL updates the authorization data with the DBW value in the "Re-authorization Request" message and sends a "Re-authorization Answer" message to the AR; if the remote re-authorization is unsuccessful, AAAL sends a "Re-authorization Answer" message including a "Failure" flag and the session key.

Reception of Timeout Events and Other Events

When the timer for the remote "Re-authorization Answer" alarms, AAAL regards the situation as the remote re-authorization is unsuccessful.

When the Session Key Timer alarms, it sends a message to MN. When the Session Key expires, AAAL removes all the cached state information for the MN and informs MAP to tear down the QoS path for the MN.

5.5 Summary

This chapter first gave a system overview including the system architecture, the involved protocols and the corresponding provided services. Then the protocol operations were given in cases of power-up, inter-domain handover and intra-domain handover.

After the description of the protocol operations, the design details of the involved protocols were given:

- enhanced advertisements and a new handover criteria;
- two-step authentication - a cookie verification a preliminary check in the First Step;
- interation of QoS and mobility signaling in one process;
- parallerizing the interated process and the re-authorization process;
- IPSec security association establishment for data protection over wireless channel;
- CASP Mobility Client Protocol as the QoS signaling;
- QoS-aware re-authorization process.

Finally the specification of the designed protocols for the cases of the “intra-domain handover” and “session update” were described by illustrating the operations of each entity. Note that in the case of “session update”, the QoS signaling should propagate from the AR to the CN since the whole path should get involved. In the specification section, however, the QoS signaling on the path only from the AR to the MAP is illustrated.

The protocol meets the requirements identified in Section 3.4 in the follow aspects:

- **Mutual authentication**:
  - authenticating MN: in inter-domain handover cases, the authentication is done with the help of AAAH; in intra-domain handover cases, the authentication is performed in two steps: cookie verification as the first step and session key check as the second step. When receiving a QoS request, an access point verifies a cookie as the first step of authentication. The verification is successful, the mobile host is regarded as a credible user. When the access point receives the session key, it performs the second step of authentication with the session key.
  - authenticating AR: it is achieved based on the trust relationship between each access router and the MAP or AAAL since only the genuine AR is able to get the session key and TmpKeyDistKey.

- **The selection of the best suited handover target**: Enhanced advertisements contain QoS information such as aggregate available information. It helps the mobile host to select the most suited access point to perform a handover;

- **IntServ support in Micro-mobility:** CASP Mobility Client Protocol is designed as a modular and light-weighted QoS signaling protocol. The CASP QUERY message contains a range of QoS parameters. We use a range of QoS parameters rather a specific value for the resource reservation so as to avoid multiple round trips for QoS negotiation;

- **DiffServ support and QoS-aware authorization:** A selected value for the first authorization. If the upper bound value were used for the first authorization, when the mobile host upgraded its QoS request later, the access network had to re-perform the authorization involving the mobile host's home domain unavoidably. Therefore, performing the first authorization with a selected value can not only solve the problem, but also avoid distributing the subscribed value of a mobile host from its home domain to the visited domain. Authorization data is available in the visited access network so that re-authorization may be done with involving the mobile node's home domain. Policy is deployed at the new access router after a successful authorization process to enable the DiffServ.

- **Optimization of QoS-aware micro-mobility:** Closely coupled QoS and mobility processes in the intra-domain handover cases. QoS signaling in the mobility signaling are integrated in CASP QUERY message. Also Parallelizing QoS and BU joint process and the re-authorization process in the intra-domain handover cases makes it possible that MN has requested service without waiting for the result of the re-authorization process.

- **Data protection over wireless channel:** An IPSec security association can be established efficiently inband with the signaling protocol between the new AR and the mobile node, in order to protect data over the wireless channel.

  In such a case, before the re-authorization process is complete, the user traffic is protected with a temporary security association.

Even though the protocol operations of the initial registration (i.e. power-up cases) and the inter-domain handovers are not within the design scope of the thesis, protocol operations of the two scenarios are required to cooperate with the operations of the intra-domain cases which are the focuses of the thesis. The protocol operations refer to Appendix B.

Chapter 6

Security Analysis of the Cryptographic Protocol against Denial of Service

In this Chapter, a cost-based approach is applied to evaluate the robustness of the cryptographic propotocol (i.e. crypto aspects of the signaling protocol) in the thesis against Denial of Service (DoS) attacks. The DoS attacks are caused by either exhausting cpu cycles of network nodes and the signaling capacity of network links or gaining access to network resources due to access control failure.

The robustness of the protocol against the resource exhaustion DoS attack will be evaluated first with a cost-based approach. Then DoS attacks due to access control failure will be discussed then. Finally, a conclusion will be given.

6.1 Resource Exhaustion DoS Attack

One of the most common Denial of Service (DoS) attacks is the resource exhaustion attack. In the context scenarios, attackers may send extensive bogus requests to an access network enforcing it to waste resources. The victim access network should take some measures to defense the attacks. (Such an access network including entities will be called as a "defender" in the following text.)

A defender suffers from a Denial of Service (DoS) attack only when the cost of the defending actions is so remarkable that the performance of the defender degrades dramatically. To analyze the robustness of a cryptographic protocol, a ratio of the cost to the defender's total available resources is more useful than an absolute value such as the CPU cycle and the storage space [60].

With the increasing of the attacking intensity (whether caused by one attacker or more), in general speaking, if the ratio can be kept low, a cryptographic protocol is judged as a secure protocol against Denial of Service (DoS).

Since the aimed security goal is to prevent a system from being broken down easily by a not-so-strong attacker, the method in judging the robustness of a protocol against Denial of Service (DoS) in the context is to compare the costs to a defender and to an attacker. That means if the cost to an attacker is trivial (from the point of view of attacker) in comparison with the cost of all the corresponding actions (from the point of view of defender), then the protocol is judged to be insecure against Denial of Service (DoS); otherwise, the protocol is judged to be secure.

Meadows introduced an effective way namely a cost-based approach in analyzing Denial of Service (DoS) [60]. She modified the fail-stop model which was first mentioned by Gong and Syverson [31] by considering the cost of all triggered by a bogus message. She defined a function Γ mapping the actions in a protocol to costs. A protocol is fail-stop if a principal can not engage in a protocol execution any longer unless an attacker pays more effort more than $\Gamma(A)$. If the $\Gamma(A)$ is trivial in comparison with the cost of all corresponding actions of a defender, the protocol is insecure.

There are some issues, however, in applying the cost-based approach in [60] to the proposed cryptographic protocol.

- How to measure the cost of an attacker and of a defender in the wireless scenario. Since messages are transmitted over the open air, the security threats are different from those in the wired network addressed in [60]. To characterize the potential attacks Is not a trivial challenge.

- the costs of different actions are not directly comparable. For example, cryptographic computation takes CPU cycles while state information takes storage space;

- The framework in [60] is applied in a simple strong authentication case. The protocol with weak authentication like the security protocol has never be analyzed with the cost-based approach.

Generally an attacker has two ways to cause a defender to waste resource: first by initiating a protocol operation and making defender to participating in the protocol run; secondly by creating and sending a bogus instance of a message to keep defender processing it.

Accordingly, the *accept events* in the protocol specification detailed in the preceding chapter will be examined to analyze the first attacking approach; the *verification events*, including all the check operations in the protocol will be examined to address the second one.

The following sections are organized as follows: the notions used for the cost based approach will be introduced first; the analysis on *accept events* and *verification events* is then carried out; final conclusive remarks are given.

6.1.1 Overview of The Cryptographic Protocol

In this section the cryptographic protocol, which has been described detailed in Chapter 5, is iterated by using the *Alice-and-Bob Specification* defined in [60]. The trust relationship is summarized shown in Figure 6.1. In addition to the long-term static relationships represented with solid line in the figure, short-term dynamic relationships are set up between MN and the access network during handover procedure.

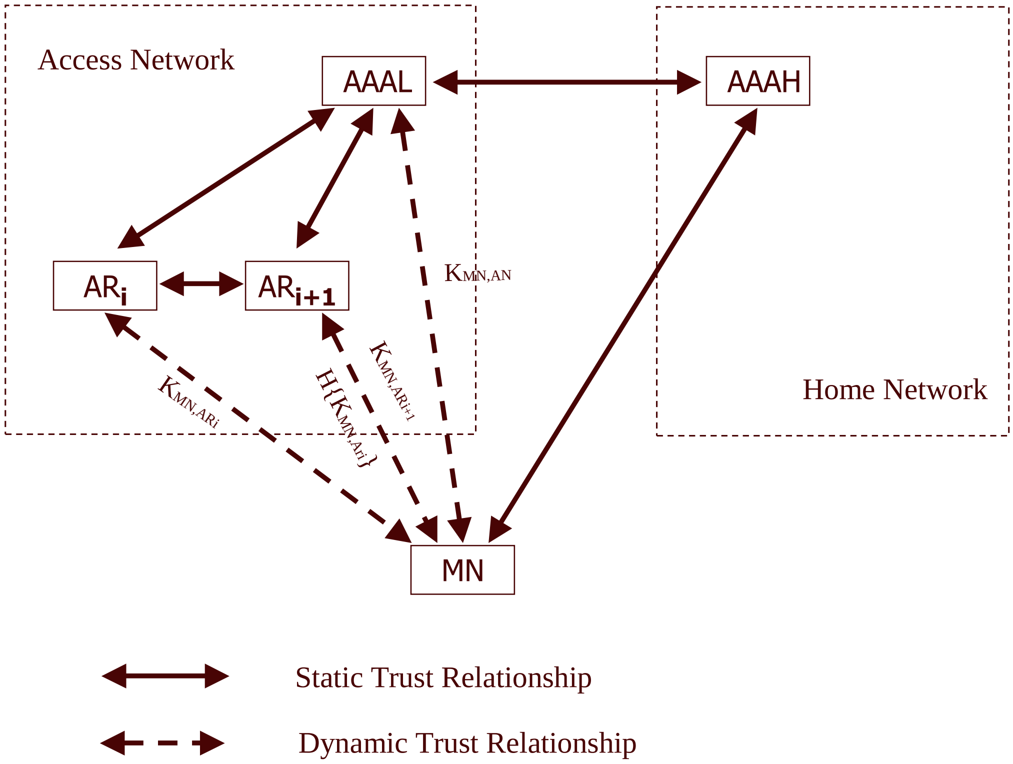


Figure 6.1: Trust Model

An annotated *Alice-and-Bob Specification* is described as the following:

$A \rightarrow B : O_1, ..., O_k \mid\mid M \mid\mid V_1, ..., V_n,$

in which O_i and V_i denote the operations of A and B respectively. M [1] responds to the message sent by A.

The operations happen in the order from left to right. That means O_i *desirably-precedes* O_j for which $i < j$; sending of message M *desirably-precedes* the receipt of a message M' for which $i < j$; V_i *desirably-precedes* V_j for which $i < j$.

Based on the *Alice-and-Bob Specification* the cryptographic protocol which is shown in Figure 6.2 can be iterated as follows:

[1] Note that the message M sent by A may not be the message received by B.

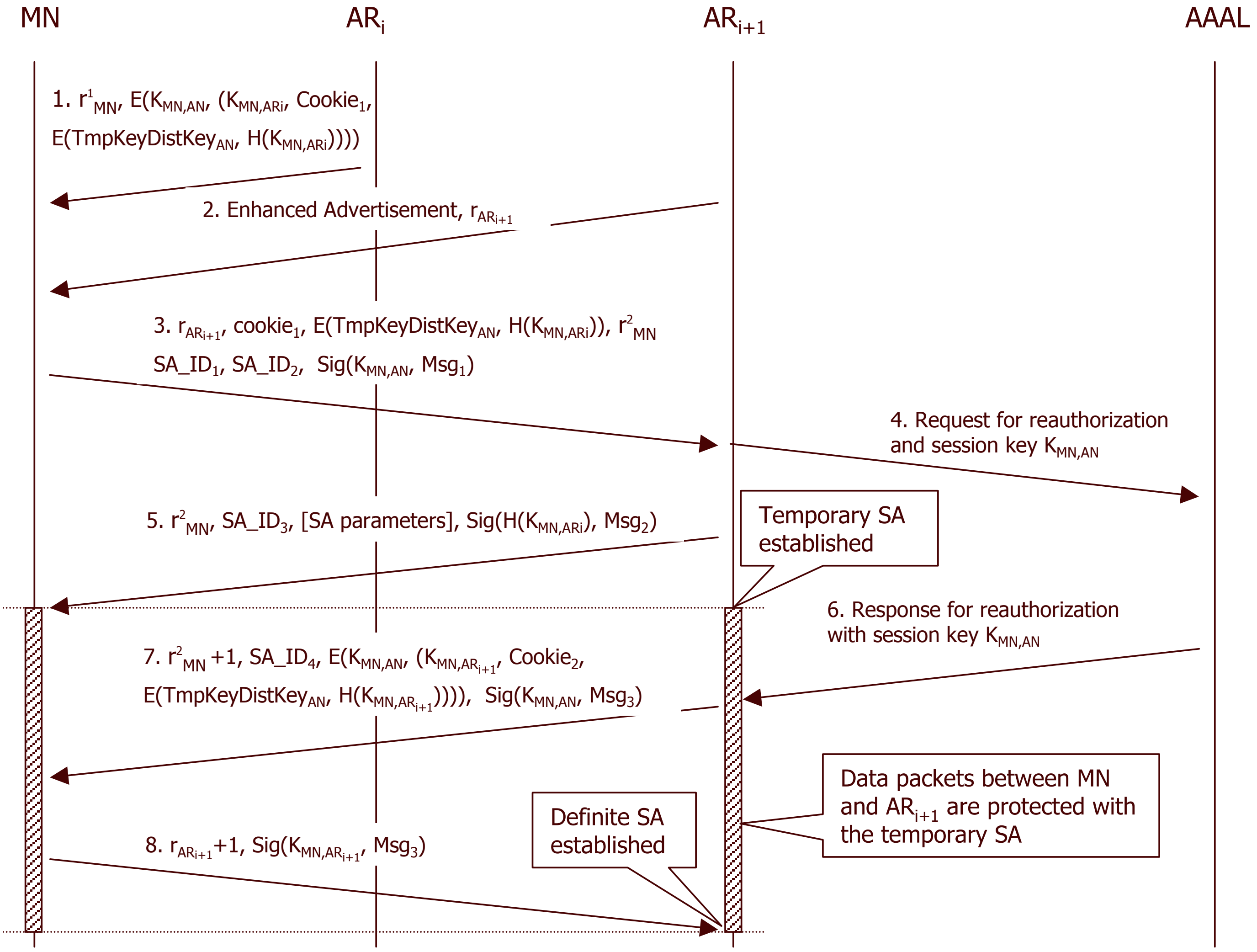

MN
AR_i
AR_{i+1}
AAAL
1. r^1_{MN}, $E(K_{MN,AN}$, $(K_{MN,ARi}$, $Cookie_1$, $E(TmpKeyDistKey_{AN}$, $H(K_{MN,ARi}))))$
2. Enhanced Advertisement, r_{ARi+1}
3. r_{ARi+1}, $cookie_1$, $E(TmpKeyDistKey_{AN}$, $H(K_{MN,ARi}))$, r^2_{MN} SA_ID_1, SA_ID_2, $Sig(K_{MN,AN}$, $Msg_1)$
4. Request for reauthorization and session key $K_{MN,AN}$
5. r^2_{MN}, SA_ID_3, [SA parameters], $Sig(H(K_{MN,ARi})$, $Msg_2)$
Temporary SA established
6. Response for reauthorization with session key $K_{MN,AN}$
7. r^2_{MN} +1, SA_ID_4, $E(K_{MN,AN}$, $(K_{MN,AR_{i+1}}$, $Cookie_2$, $E(TmpKeyDistKey_{AN}$, $H(K_{MN,AR_{i+1}}))))$, $Sig(K_{MN,AN}$, $Msg_3)$
Data packets between MN and AR_{i+1} are protected with the temporary SA
Definite SA established
8. r_{ARi+1}+1, $Sig(K_{MN,AR_{i+1}}$, $Msg_3)$

Figure 6.2: Crypto Aspects of the Protocol

1. $AR_i \rightarrow MN : encrypt_1 \;||$ r^1_{MN}, $E(K_{MN,AN}$,$(K_{MN,AR_i}$, $cookie_1$, $E(TmpKeyDistKey_{AN}$,$H(K_{MN,AR_i}))))$ $||\; retrievenonce_1$, $checkname_1$, $decrypt_1$, $accept_1$, $storeinfo_1$

   This message is the registration answer after the Mobile Node (MN) registers successfully in the access network in an inter-domain handover or a power-up case.

   The cryptographic operation $encrypt1$ taken by AR_i includes the following actions:

   - generate a key K_{MN,AR_i};
   - do a hash function on the key: $H(K_{MN,AR_i})$;
   - encrypt the hash code with the key $TmpKeyDistKey_{AN}$ [2];

[2] key is distributed periodically from the local AAA server to every Access Router (AR)

- generate a cookie [3];
- encrypt three pieces of information with the session key $K_{MN,AN}$ [4] [5]: the generated key K_{MN,AR_i}, the $cookie_1$ and the encrypted hash code;

Then AR_i sends the above encrypted information along with a nonce r_{MN_1} to MN. The operations of MN after receiving the message are

- retrieve the nonce r^1_{MN} and $checkname_1$ to verify whether it is the correct nonce;
- decrypt the information with the session key $K_{MN,AN}$ [6];
- if MN decrypts the message successfully, it accepts the message based on the following reason: since MN can decrypt successfully the message with the session key, it is proved that the message must be encrypted by an entity who can access the session key. Upon believing in the static trust relationship between the access network and the home network and the static trust relationship between the AAAL and ARs, MN believes that the message is not bogus;
- after accepting the message, MN stores the information: $storeinfo_1$.

2. $AR_{i+1} \rightarrow MN : storenonce_1 \mid\mid Enhanced\_Advertisment, r_{AR_{i+1}} \mid\mid accept_2$

   When MN receives enhanced advertisements from new Access Router (AR)s, after experiencing the handover decision process with the relevant information contained in the enhanced advertisements, MN may decide to move to a new AR. Additionally, each advertisement also contains a corresponding random number $r_{AR_{i+1}}$. The only operation a AR needs to do is to cache a nonce $storenonce_1$.

   Authenticating advertisements is out of scope of the thesis [7]. Since it is assumed that an attacker is not capable to fake advertisements being sent by the access network, MN simply accept them receiving advertisements.

3. $MN \rightarrow AR_{i+1} : storenonce_2, sign_1 \mid\mid$
   $r_{AR_{i+1}}, cookie_1, E(TmpKeyDistKey_{AN}, H(K_{MN,AR_i})),$
   $r^2_{MN}, SA\_ID_1, SA\_ID_2, Sig(K_{MN,AN}, Msg_1)$
   $\mid\mid retrievenonce_2, checkname_2, checkcookie_1, accept_3, decrypt_2, storeinfo_2,$
   $notification\_message, mobility\_singaling\_message$

[3] how generate a cookie refers to the design and specification chapter

[4] the session key is distributed from the Mobile Node (MN)'s home network to the access network

[5] $E(K, X)$ denotes encryption of X with key K

[6] MN is able to derive the session key based on the long-term trust relationship with its home network

[7] TESLA [76] and its extensions may be useful for the purpose

In case MN decides to move $r_{AR_{i+1}}$, it first caches a nonce - r^2_{MN}. Then it signs the following information with the session key $K_{MN,AN}$:

- $r_{AR_{i+1}}, cookie_1, E(TmpKeyDistKey_{AN}, H(K_{MN,AR_i}))$ which were obtained from registration answer messages sent by AR_i;
- r^2_{MN}: the nonce stored by MN;
- SA_ID_1, SA_ID_2 which serve as $SPI1$ and $SPI2$ for the temporary *IPSec* Security Association (SA) and the definite one respectively in the direction from MN to AR_{i+1};

When receiving a registration request message, AR_{i+1} first checks whether the nonce $r_{AR_{i+1}}$ was issued by it. Afterwards, it performs a cookie check - $checkcookie_1$ [8]. If the cookie verification is successful, AR_{i+1} accepts the message. Then it decrypts to obtain $H(K_{MN,AR_i})$ by using the key $TmpKeyDistKey_{AN}$.

It also caches the information $r_{MN_2}, SA\_ID_1, SA\_ID_2, Sig(K_{MN,AN}, Msg_1)$. Msg_1 refers to the registration request message. $Sig(K_{MN,AN}, Msg_1)$ will be used to authenticate the request by proving the possession of the session key when AR_{i+1} receives the session key $K_{MN,AN}$ from Local AAA Server (AAAL). As AR_{i+1} accepts the request, AR_{i+1} sends a notification message to other Access Router (AR)s about the use of the cookie, and initiates the QoS-embedded mobility process. Meanwhile, it also starts the reauthorization process.

4. $AR_{i+1} \rightarrow AAAL : encrypt_2 \mid\mid Request\_for\_reauthorization,$
   $Request\_for\_K_{MN,AN} \mid\mid decrypt_3, accept_4$

   In the reauthorization process, AR_{i+1} sends a message to Local AAA Server (AAAL) requesting for reauthorization service and the session key $K_{MN,AN}$. The communication between AR_{i+1} and AAAL is protected under the static Security Association (SA) between them.

   Suppose this process takes longer time than the QoS-embedded mobility process due to various reasons such as involving Home AAA Server (AAAH) for the reauthorization. Therefore, AR_{i+1} receives the acknowledgement message in the mobility process.

5. $AR_{i+1} \rightarrow MN : sign_2 \mid\mid r^2_{MN}, SA\_ID_3, [SAParameters],$
   $Sig(H(K_{MN,AR_i}), Msg_2) \mid\mid retrievenonce_3, checkname_3, checksig_1,$
   $accept_5, storeinfo_3$

[8] the cookie verification was detailed in the protocol design and specification chapter

AR_{i+1} signs the information r_{MN_2}, SA_ID_3, $[SAParameters]$ with key $H(K_{MN,AR_i})$. SA_ID_3 indicates the SPI for the temporary IPSec Security Association (SA) in the direction from AR_{i+1} to MN. $[SAParameters]$ contains the necessary parameters for the IPSec SA set up such as the cryptographic algorithm to be used.

Since MN possesses the key $H(K_{MN,AR_i})$, it can check the signature with the key - $checksig_1$ after verifying the nonce. If the signature check passes, it accepts the message. Then MN caches the related information and accept the message since only the legitimate entities in the access network can access the key $TmpKeyDistKey_{AN}$ so as to derive the correct $H(K_{MN,AR_i}$, with which to sign the message.

After MN accepts the message, MN and AR_{i+1} start using the temporary IPSec SA to protect the user data communicating between them.

6. $AAAL \rightarrow AR_{i+1} : encrypt_3 \,||\, Reauthorization\_Result\_with\_K_{MN,AN}$
$||\, decrypt_4, accept_6$

   AAAL sends the reauthorization result to AR_{i+1} along with the session key $K_{MN,AN}$. This message is also protected under the static Security Association (SA) between them.

7. $AR_{i+1} \rightarrow MN : checksig_2, encrypt_4, sign_3 \,||$
$r_{MN}^2 + 1, SA\_ID_4, E(K_{MN,AN}, (K_{MN,AR_{i+1}}, cookie_2,$
$E(TmpKeyDistKey_{AN}, H(K_{MN,AR_{i+1}})))), Sig(K_{MN,AN}, Msg_3)$
$||\, retrievenonce_4, checkname_4, checksig_3, accept_7, decrypt_5, storeinfo_4$

   In case of a successful reauthorization, AR_{i+1} authenticates the registration request with the session key $K_{MN,AN}$: check the signature of $Sig(K_{MN,AN}, Msg_1)$. If the authentication is successful, as described in Step 1, AR_{i+1} performs $encrypt_4$ which includes preparing a new temporary key $E(TmpKeyDistKey_{AN}, H(K_{MN,AR_{i+1}}))$, generating a new cookie $cookie_2$, encrypting them along with the key $K_{MN,AR_{i+1}}$ with the session key $K_{MN,AN}$.

   Then AR_{i+1} sends the following to MN:

   - the encrypted information;
   - the MN's nonce plus 1;
   - the SPI SA_ID_4 which indicates the SPI for a definite IPSec SA in the direction from AR_{i+1} to MN;
   - a hash code of the above information. The hash code, which serves as a signature, is generated with the session key.

When MN receives the message, it performs the following operations:

- retrieve the nonce $r^2_{MN}+1$ and verify whether it is the correct nonce;
- check signature with the session key $K_{MN,AN}$;
- accept the message if the check passes;
- decrypt the information also with the session key $K_{MN,AN}$;
- after decrypting successfully, MN caches the information $K_{MN,AR_{i+1}}$, $cookie_2$, and $E(TmpKeyDistKey_{AN}, H(K_{MN,AR_{i+1}}))))$.

MN accepts the message based on the assumption that the session key shared with the access network has not been compromised.

8. $MN \rightarrow AR_{i+1} : sign_4 \mid\mid r_{AR_{i+1}} + 1, Sig(K_{MN,AR_{i+1}}, Msg_4)$
   $\mid\mid retrievenonce_5, checkname_5, checksig_4, accept_8$

   When MN receives the message for setting up a definite SA, it sends an acknowledgement message to AR_{i+1} which includes a nonce $r_{AR_{i+1}}+1$ and a signature with the key shared between MN and AR_{i+1}: $K_{MN,AR_{i+1}}$.

   After verifying the nonce and the signature successfully, AR_{i+1} accepts the message. The MN and AR_{i+1} substitute the temporary IPSec SA with the new SA for their data protection.

6.1.2 Introduction on the Cost Functions and Tolerance Relation

- Cost Sets and Protocol Cost Functions

  According to a definition in [60], in an annotated *Alice-and-Bob Specification*

  $A \rightarrow B : O_1, ..., O_k \mid\mid M \mid\mid V_1, ..., V_n$,

  the cost function δ' of each *verification event* V_j is

  $\delta'(V_j) = \delta(V_1) + ... + \delta(V_j)$

  The cost function of a *verification event* is the sum of the costs of all the individual event from the receipt of the message to (and including) a verification event.

  the cost function of an *accept event* $\Delta(V_n)$ is the sum of all the costs of all the B's operations *desirably-preceding* V_n, plus the sum of the costs of all B's operations of *creating and sending* the next messages, if any, which result from the *accepting* of this message M.

  $\Delta(V_n) = \Delta(V_1) + ... + \Delta(V_{n-1}) + \Delta(\text{resulted actions from } V_n)$

  For example, the cost function of accepting the registration request message from MN includes not only the cost of cookie verification, but also the costs of creating and sending of the cookie notification message, QoS-embedded mobility signaling message and reauthorization message because these three messages are the direct results of accepting of the registration request message.

A simplest cost set has been proposed in [60],which is $expensive > medium > cheap > 0$. Exponential cryptographic operations such as *exp, checksig, and sign* are expensive; other cryptographic operations such as *encrypt, decrypt, preexp* are medium and all other events are cheap.

There are two problems with the definition:

- the relationships among the four difficulty levels are not clear;
- the same operation such as *encrypt or decrypt* should may not be mapped to the same difficulty level at different entities such as AR and MN.

In the following analysis, the relationships of difficulty levels will treated as variables; and the difficulty level of operations at different entities will be differentiated.

- Attacker Cost Functions and Tolerance Relation

  First of all, the attacker's capabilities and the corresponding cost functions should be determined such as reading a message over the air interface may cost *medium*.

  In the attacker cost set, two new difficulty levels are introduced in addition to the protocol cost set: *very expensive* and *maximal*. *maximal* is reserved for events such as cryptanalysis to break a one-way hash function, which is assumed to be infeasible.

  The same problems also exist: the definitions of the cost sets are not quantified clearly; the meaning of an attacker's *medium* cost may be different from that of MN.

  The notion of *tolerance relation* is used for evaluating whether or not a cryptographic protocol is secure against Denial of Service (DoS). *tolerance relation* is defined as a pair of a defender's cost and a attacker's cost c, g that a situation is tolerated in which an attacker can not make a defender to expend resources of cost *c* or greater without expending its resources of cost *g* or greater [60].

  For an actual cost pair c', g', if $c' \leq c$ *AND* $g' \geq g$ is true, c', g' is within the *tolerance relation*. The notion *tolerance relation* indicates the tolerable relationship between an estimate of an attacker's resource committed to an attack and an estimate of the defender's available resource: how much defender's resource can be provided for a certain level of security; how much insecurity is acceptable for a certain amount of cost.

Table 6.1: Cost Functions

Entity	Operation	Cost [60]
MN	checkname	$cheap_{mn}$
	compute hash code	$cheap_{mn}$
	encrypt	$medium_{mn}$
	decrypt	$medium_{mn}$
AR and AAAL	checkname	$cheap_{ar}$
	compute hash code	$cheap_{ar}$
	encrypt	$medium_{ar}$
	decrypt	$medium_{ar}$
Attacker	read nonce	$medium_{attr}$
	break an encrypted code	$maximal_{attr}$
	break a hash function	$maximal_{attr}$

The notion of *tolerance relation* is used to evaluate the proposed cryptographic protocol illustrated in Section 6.1.1 in the following subsections.

6.1.3 Analysis on the Accept Events

For each *accept event* E, the cost pair $\Delta(E), \Theta(E)$ will be computed, where $\Delta(E)$ is the protocol cost function of *accept event* E and $\Theta(E)$ presents the attack cost function to make the *accept event* E occur.

The cost pair will be compared with predefined *tolerance relation* to determine whether the step of the protocol run is secure.

Based on the protocol cost function and attack cost function shown in Table 6.1, the analysis on the *accept event*s are described as follows:

- $accept_1$:

  $\Delta(accept_1)$ is the sum of the costs of all the operations *desirably preceding* $accept_1$. When the dominating factors which are $checkname_1$ and $decrypt_1$ are taken into account,

  $\Delta(accept_1) = cheap_{mn} + medium_{mn}$

  $\Theta(accept_1)$ is the attack costs of all the actions to make the $accept_1$ occur at MN. The actions and the corresponding costs are

  - reading MN's first registration message to obtain the nonce r^1_{MN} (cost = $medium_{attr}$);
  - performing cryptanalysis to get the session key $K_{MN,AN}$ (cost = $maximal_{attr}$).

Therefore,

$\Theta(accept_1) = maximal_{attr}$

- $accept_2$:

  $\Delta(accept_2)$ can be determined mainly by the operations resulting from the accepting the advertisement: the sum of the costs of the operations of making handover decision (cost = $cheap_{mn}$), signing the re-registration message with the session key (i.e. doing a hash function, cost = $cheap_{mn}$). Therefore,

  $\Delta(accept_2) = 2 * cheap_{mn}$

  Since it is assumed that attacker is not capable to impersonate an AR to emit enhanced advertisement,

  $\Theta(accept_2) = maximal_{attr}$

- $accept_3$:

  Since AR_{i+1} needs to check the nonce (cost = $cheap_{mn}$), perform a cookie verification (i.e. do a hash function) (cost = $cheap_{mn}$) before accepting the message, and also performing $decrypt_2$ (cost = $medium_{mn}$), sending a notification message to the cookie generator AR (cost = $medium_{mn}$), initiating mobility signaling process (cost = $medium_{mn}$) and an reauthorization process (cost = $medium_{mn}$) are also the results of the accepting of the message, based on the cost message,

  $\Delta(accept_2) = 2 * cheap_{mn} + 4 * medium_{mn}$

  In order to generate an acceptable registration message, an attacker needs to

  - obtain the nonce from the enhanced advertisement (cost = medium);
  - generate a cookie with the correct cookie key (cost = maximal);
  - sign the message with the session key (cost = maximal).

  Therefore, the task to generate an acceptable registration message with the acknowledgement of the cookie key and the session key is infeasible for an attacker.

  $\Theta(accept_3) = maximal_{attr}$

  However, an attacker may eavesdrop an registration message (which includes the an correct cookie and signature) and replay it to some access routers to gain access. When the attacker has gained the access, it can send bogus messages to reserve resource, causing a DoS problem. Since this kind of DoS results from the access control failure, the threats will be discussed in the next subsection.

- $accept_4$:

Under the static Security Association (SA) between the AR_{i+1} and AAAL, AR_{i+1} sends a message request AAAL to perform reauthorization and distribute the session key. Therefore, the cost of AAAL's operations mainly comes from *decrypt* whose cost is *medium*.

$\Delta(accept_4) = medium_{aaal}$

To fake an acceptable message to AAAL, an attacker needs to break the static Security Association (SA) between the AR_{i+1} and AAAL, which is assumed to be *maximal*.

$\Theta(accept_4) = maximal_{attr}$

- $accept_5$:

  To accept the message in Step 5, MN needs to check the nonce and check the signature with $H(K_{MN,AR_i})$ (i.e. perform a hash function). According to the cost function table 6.1,

  $\Delta(accept_5) = 2 * cheap_{mn}$

  In order to fake an acceptable message to MN, an attacker has to get the nonce r^2_{mn} and break the one-way hash code $E(TmpKeyDistKey_{AN}, H(K_{MN,AR_i}))$ to get the key $H(K_{MN,AR_i})$. Therefore,

  $\Theta(accept_5) = maximal_{attr}$

- $accept_6$:

  Similar to $accept_4$,

  $\Delta(accept_6) = medium_{ar}$

  $\Theta(accept_6) = maximal_{attr}$

- $accept_7$:

  The operations resulting to $accept_7$ and resulted from $accept_7$ include checking nonce (cost = $cheap_{mn}$), checking a signature by doing a hash function (cost = $cheap_{mn}$), performing $decrypt_5$ (cost = $medium_{mn}$)and signing a message in Step 8 by doing a hash function (cost = $cheap_{mn}$). Therefore,

  $\Delta(accept_7) = 3 * cheap_{mn} + medium_{mn}$

  Since an attacker needs to read a message to obtain nonce r^2_{mn}, and break the one-way hash function to get the session key,

  $\Theta(accept_7) = medium_{attr} + maximal_{attr}$

- $accept_8$

Since AR_{i+1} checks the nonce (cost = $cheap_{ar}$), also checks the signature (cost = $cheap_{ar}$),

$\Delta(accept_8) = 2 * cheap_{ar}$

The *accept event*cost to an attacker is

$\Theta(accept_8) = medium_{attr} + maximal_{attr}$

6.1.4 Analysis on the Verification Events

In an annotated *Alice-and-Bob Specification*, supposed E_1 is an event *immediately preceding* a *verification event* E_2, the cost pair ($\delta'(E_2)$, $\Theta(E_1)$) is computed.

This method can be used to justify the robustness of the cryptographic protocol against Denial of Service (DoS) in which an attacker causes a defender to waste resources processing a message before it discovers the attack.

Based on the protocol cost function and attack cost function shown in Table 6.1,the analysis on the *verification event*s are described as follows:

- Step 1:

  In Step 1, the cost of the verification operation $decrypt_1$ can be expressed as

  $\delta'(decrypt_1)$ = $\delta(checkname_1) + \delta(decrypt_1)$ = $cheap_{mn} + medium_{mn}$

  The event *immediately preceding* $decrypt_1$ is $checkname_1$. The attack cost function $\Theta(chackname_1)$ mainly consists of the costs of reading a message to get nonce r^1_{mn}, whose cost is $medium_{attr}$.

  $\Theta(chackname_1) = medium_{attr}$

  Since the only intention of the attacker is to cause MN to perform a *decrypt* and it does not care about the decryption result, after reading the nonce, the attacker needs only to fill in the field, which MN will perform *decrypt* to, with random numbers.

  When the attacker commits its effort to $medium_{attr}$, MN has to pay the cost of $cheap_{mn} + medium_{mn}$.

- Step 2:

  Even though there is no check on the received advertisements, attackers are assumed not to be capable to impersonate an AR to beacon advertisements. Therefore, this step is secure against Denial of Service (DoS).

- Step 3:

  The protocol costs of the two *verification event*s (i.e. $\delta'(checkname_2)$ and $\delta'(checkcookie_1)$) are $cheap_{mn}$.

$\delta(checkname_2, checkcookie_1)$ = 2 $*$ $cheap_{ar}$

An attacker needs to read the message (cost = $medium_{attr}$) to get the nonce r^2_{mn} and necessary cookie information because it may want to take some care to pass checks for obviously bogus values [60]. Therefore,

$\Theta(checkname_2)$ = $medium_{attr}$

- Step 4:

  The communication is protected with the standard IPSec SA. The analysis on such an approach is out of scope of the thesis. It is assumed that this step is secure.

- Step 5:

  The *verification event*s consists of $checkname_3$ and $checksig_1$. According to the cost function table 6.1,

  $\delta(checkname_3, checksig_1)$ = 2 $*$ $cheap_{mn}$

  Since an attacker also needs to read message (cost = $medium_{attr}$) to obtain the nonce r^2_{mn}, the cost pair is ($\delta(checkname_3, checksig_1)$, $\Theta(chackname_3)$).

- Step 6:

  Based on the same argument as in $Step4$, this step is assumed to be secure.

- Step 7:

  Similar to $Step5$, the *verification event*s consists of $checkname_4$ and $checksig_3$. Therefore, the protocol cost function and attack cost function are

  $\delta(checkname_4, checksig_3)$ = 2 $*$ $cheap_{mn}$;

  $\Theta(chackname_4) = medium_{attr}$ respectively.

- Step 8:

  Following the same reasoning in $Step7$, the cost functions are

  $\delta(checkname_5, checksig_4)$ = 2 $*$ $cheap_{mn}$, and

  $\Theta(chackname_5) = medium_{attr}$.

6.1.5 Conclusive Remarks

The cost pairs of *accept event*s and *verification event*s are summarized in Table 6.2 and Table 6.3 respectively.

Since the attack cost is $maximal_{attr}$ in all steps of the analysis on the *accept event*s, it can be concluded that, for any the *tolerance relation*, it is infeasible for an attacker to

Table 6.2: Summary of Analysis on *ACCEPT EVENT*

Step	Defender	$\Delta(E)$	$\Theta(E)$
1	MN	$cheap_{mn} + medium_{mn}$	$maximal_{attr}$
2	MN	$2 * cheap_{mn}$	$maximal_{attr}$
3	AR_{i+1}	$2 * cheap_{mn} + 4 * medium_{mn}$	$maximal_{attr}$
4	AAAL	$medium_{aaal}$	$maximal_{attr}$
5	MN	$2 * cheap_{mn}$	$maximal_{attr}$
6	AR_{i+1}	$medium_{ar}$	$maximal_{attr}$
7	MN	$3 * cheap_{mn} + medium_{mn}$	$maximal_{attr}$
8	AR_{i+1}	$2 * cheap_{ar}$	$maximal_{attr}$

Table 6.3: Summary of Analysis on *VERIFICATION EVENT*

Step	Defender	$\delta'(E_2)$	$\Theta(E_1)$
1	MN	$cheap_{mn} + medium_{mn}$	$medium_{attr}$
2	MN	0	$maximal_{attr}$
3	AR_{i+1}	$2 * cheap_{ar}$	$medium_{attr}$
4	AAAL	$-$	$-$
5	MN	$2 * cheap_{mn}$	$medium_{attr}$
6	AR_{i+1}	-	-
7	MN	$2 * cheap_{mn}$	$medium_{attr}$
8	AR_{i+1}	$2 * cheap_{ar}$	$medium_{attr}$

cause a defender to waste resources engaging in a protocol run up, receiving a particular message and responding to it.

If an attacker increases its power so that it can change a $maximal_{attr}$ costed task to a $expensive_{attr}$ task, according to the *tolerance relation* ($expensive_{ar}, expensive_{attr}$), Step 3 may make AR_{i+1} insecure when so many *medium* tasks mean *expensive* to it. Therefore, the cryptographic protocol is not claimed to be completely free from Denial of Service (DoS).

Concerning the *verification event*s, compared to the *tolerance relation*

$$(medium_{ar}, medium_{attr}) \;\; or \;\; (medium_{mn}, medium_{attr}),$$

the cost pair ($cheap_{mn}$ + $medium_{mn}$, $medium_{attr}$) in Step 1 shows that MN is not secure.

Actually MN has to perform *decrypt* as a verification check which means a *medium* costed task when it receives the first registration answer message. An attacker may cause Denial of Service (DoS) to MN.

Moreover, when an attacker becomes more powerful and sends bogus messages more intensively to flood the access network in Step 3, AR_{i+1} may have some trouble with the Denial of Service (DoS) attack because the protocol cost in Step 3 (i.e. $2 * cheap_{ar}$) is not free. This point will be proved in the mathematical analysis chapter.

6.2 DoS Attack due to Access Control Failure

During the transmission of the re-registration message in the air as shown in Step 3, an attacker can copy the message and replay it to ARs which are destined to accept the cookie contained in the message. If the relay of the message arrives at the ARs earlier than the notification message does, the ARs will *accept* the message due to the success of the $checkname_2$ and $checkcookie_1$. Thus the attacker passes the first authentication, and makes the ARs to commit its resources on the tasks such as sending notification messages with the protection of the static SA between two ARs, initiating the joint resource reservation and mobility process, and starting the reauthorization process.

This kind of DoS (i.e. wasting resources and marking resources as occupied) due to the access control failure is different from the resource exhaustion DoS discussed in the preceding subsection. In this subsection, the cost of this kind of DoS is discussed.

6.2.1 Analytic Network Architecture

An hexagonal cellular HMIPv6 network architecture is assumed in the analysis as shown in Figure 6.3.

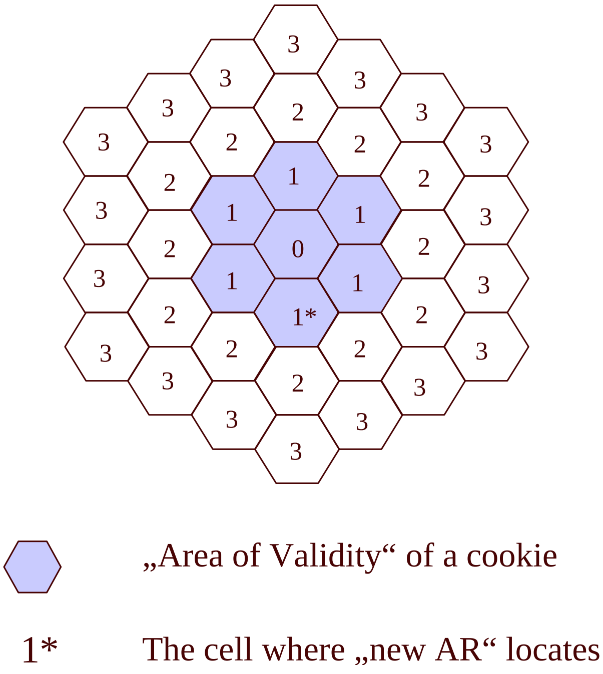

0
1
1*
2
3
„Area of Validity“ of a cookie
1* The cell where „new AR“ locates

Figure 6.3: Hexagonal Cellular HMIPv6 Network Architecture

A MAP domain is assumed to cover rings of cells. One ring r^{th} $(r \geq 0)$ consists of $6r$ cells. The total number of cells in a MAP domain is calculated as

$$\sum_{r=1}^{R} 6r + 1 = 3R(R+1) + 1 \tag{6.1}$$

where *R* is the number of rings a MAP domain covers.

Each cell has an *Access Router (AR)*. A cookie generator is assumed to locate in the cell numbered as “0”. The *Area of Validity (AOV)* of a cookie is assumed to be the area centered by the cell “0”. The *AOV* can be the scope edged by the ring $1, 2, ..., i$. Therefore, the number of cells of the *AOV* can also be obtained from Eq. 6.1 as

$$3i(i+1) + 1 \tag{6.2}$$

Figure 6.3 shows the *AOV* which is edged by the ring 1 in the shaded part. The cell numbered as 1* is assumed to be the cell where the “new AR” (i.e. to which a cookie is presented) locates.

The involved entities are shown in Figure 6.4. An attacker may eavesdrop the cookie from the request message from an legitimate mobile user, and replay it to either the “new AR (NAR)” or other ARs in the *AOV*. It is assumed that there are two hops (i.e. four Ethernet interfaces) from an AR to the AAAL server.

The notations used the analysis are shown in Table 6.4.

Table 6.4: Notations In the Analysis

Symbol	Remark
w_{EN}	Weight of an “Encryption” operation
w_{DE}	Weight of an “Decryption” operation
w_c	Weight of the transmission over an Ethernet interface
w_w	Weight of the transmission over a wireless interface
$w_{session\_key}$	Weight of the session key lookup at AAAL
w_{AUTH}	Weight of an authentication operation using a session key at AR
w_{co}	Weight of a cookie verification at AR
w_{att}	Weight of attacker’s generating a bogus request, including parsing the genuine message and composing a bogus message
r_{cc}	Cell crossing rate per mobile. $r_{cc} = \eta \frac{v}{k}$
v	velocity of a mobile host
k	radius of a cell
η	constant proportional factor of relation between velocity and cell radius
m	number of mobiles in one cell. $m = \delta \frac{3}{2}\sqrt{3}k^2$
δ	density of mobiles in the handover domain
α	percentage of cookies in genuine requests being eavesdropped and replayed
β	possibility of the replayed cookies being accepted by an AR
γ	percentage of cells under attack
i	*i*th ring forming the boundary of *AOV*
R	number of rings covered by a MAP domain

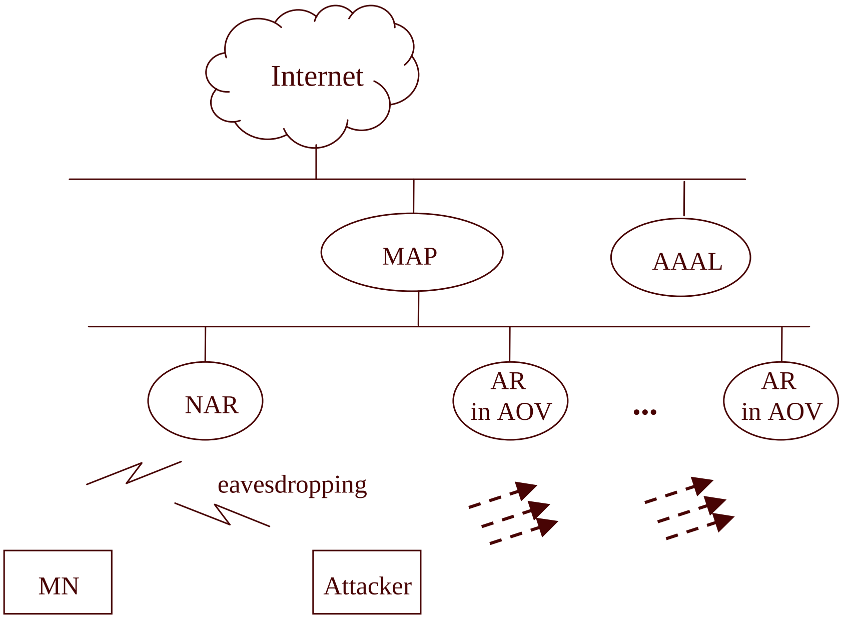


Figure 6.4: Involved Entities in the Analytic Network Architecture

6.2.2 Operation Costs of The Cookie Mechanism

The analysis on the operation costs of the cookie mechanism is classified into two scenarios: 1.) the replay of a cookie is sent to the “new AR” to which the original cookie is presented; and 2.) the replay of a cookie is sent to the ARs in the *AOV* with the exception of the “new AR”.

Scenario 1

When the replay of a cookie arrives to the “new AR” before the genuine cookie, the “new AR” reserves resources to the bogus request until it performs the second-step authentication with the session key (if any) transmitted from AAAL. The maximum time of the period consists of encrypting and sending request for the session key, decrypting the message and looking for the session key, encrypting and sending a reply message, decrypting the reply and authenticating the request. It can be calculated as

$$w_{EN} + 4w_c + w_{DE} + w_{session\_key} + w_{EN} + 4w_c + w_{DE} + w_{AUTH} \tag{6.3}$$

“new AR” may reject the legitimate request with the genuine cookie since it has accepted the replayed one.

Scenario 2

When an attacker eavesdrops a cookie and replays it to the ARs in the *AOV* except for the “new AR”, the replay may arrive earlier than the notification message. As shown in

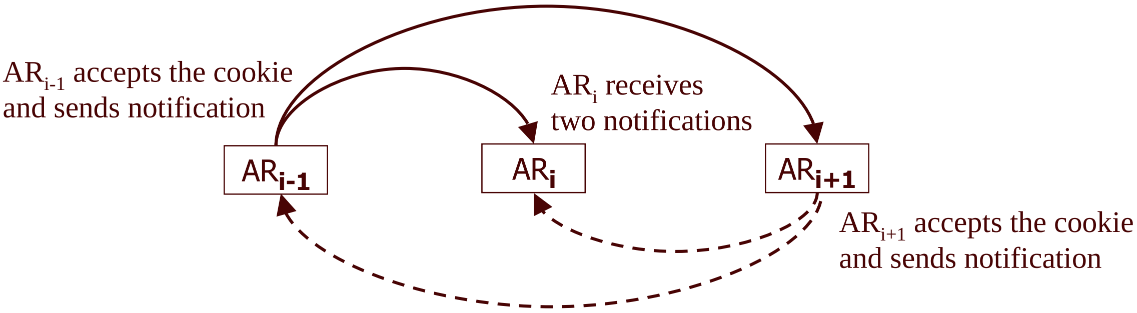

AR$_{i-1}$ accepts the cookie and sends notification
AR$_i$ receives two notifications
AR$_{i-1}$
AR$_i$
AR$_{i+1}$
AR$_{i+1}$ accepts the cookie and sends notification

Figure 6.5: An Example Of a Successful Replay of a Cookie From an Attacker

Figure 6.5 where it is assumed that AR_i is the cookie generator, and AR_{i+1} is the "new AR" whereas AR_{i-1} is the AR which accepts the replayed cookie. Since both AR_{i-1} and AR_{i+1} accept the cookie, they send the notification in the *AOV*.

The operations of an AR when receiving a notification message are shown in Figure 6.6. First the AR checks whether the notified cookie has been on its "notified cookie" list. If no, the AR adds it on the list; if yes, the AR can ensure that the *replay* occurs. If the cookie is not the cookie where has been accepted by itself, the AR knows that the *replay* occurs at other ARs; otherwise, the *replay* may have occurred at either this AR or the sender of the notification message. The AR holds the session traffic until the second-step authentication finishes. If the authentication passes, AR knows that the *replay* has occurred at the sender of the notification message and the cookie previously accepted by itself is a genuine one. Therefore, it continue the session. If the authentication fails, the AR knows that the cookie previously accepted by itself is a replayed one. It sends a message to tear down the QoS path for the session. The duration for the false reservation can also be determined by Eq. 6.3.

The operation costs of genuine requests and bogus requests are discussed as follows.

- The Operation Costs of Genuine Requests

  The operation costs of a *sending AR* (i.e. the sender of notification messages) can be expressed as

$$C_{cookie\_t\_sender} = mr_{cc}(w_w + w_{co}) + mr_{cc}(w_{EN} + w_c)3i(i+1) \qquad (6.4)$$

  mr_{cc} presents how many requests in one cell. For each of the requests, the cost of the wireless interface w_w and the cost of the cookie verification w_{co} are required. After the cookie verification passes, notification messages should be sent to all the ARs in the *AOV* except the "new AR" corresponding to each of the requests. The number of the notification messages can be derived from Eq. 6.2, which is $3i(i+1)$. The costs of sending a notification message consist of the cost of encrypting the message w_{EN} provided that each pair of ARs communicate with

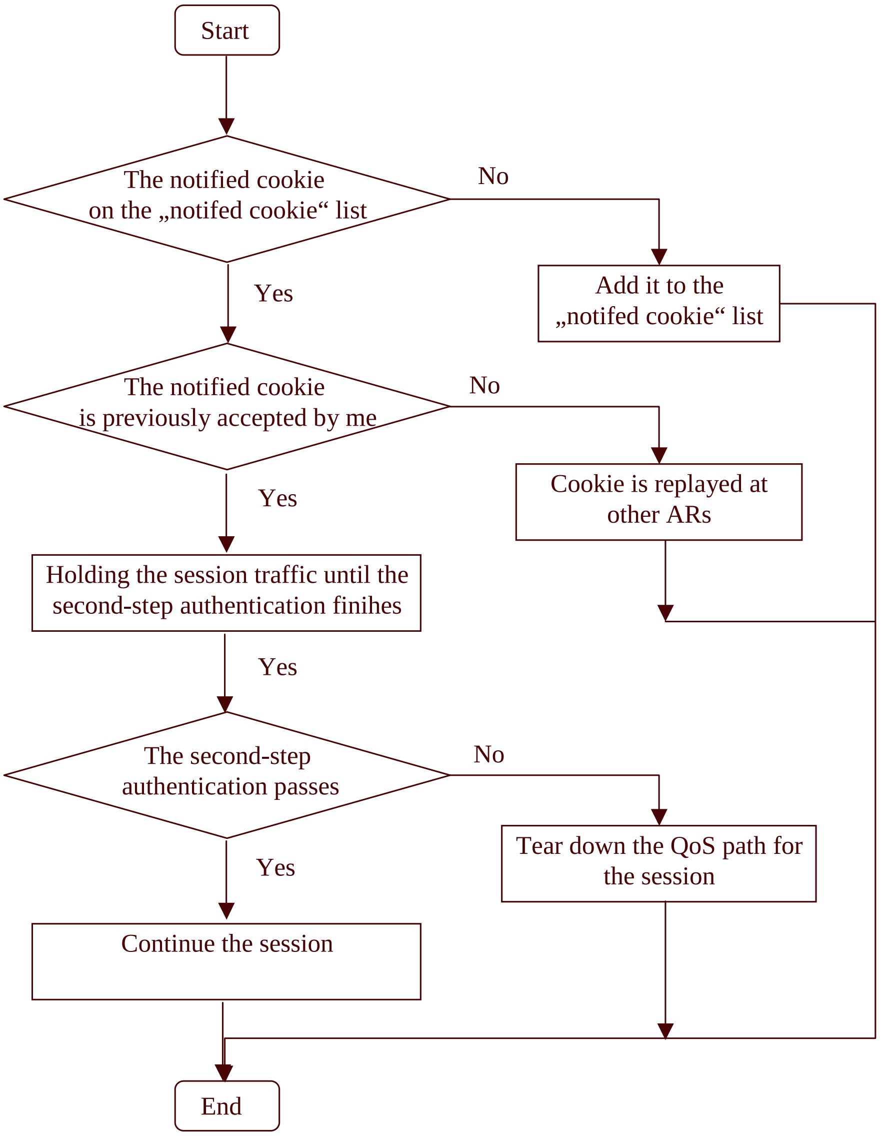

Start
The notified cookie on the „notifed cookie“ list
No
Add it to the „notifed cookie“ list
Yes
The notified cookie is previously accepted by me
No
Cookie is replayed at other ARs
Yes
Holding the session traffic until the second-step authentication finihes
Yes
The second-step authentication passes
No
Tear down the QoS path for the session
Yes
Continue the session
End

Figure 6.6: Operations of an AR When Receiving a Notification

protection of a static Security Association (SA), and the cost of transmitting the message over an Ethernet interface provided that an AR is one hop away from another.

The operation costs of an *receiving AR* which receives notification message can defined as

$$C_{cookie\_t\_receiver} = 3i(i+1)mr_{cc}(w_c + w_{DE}) \qquad (6.5)$$

An AR receives the notification messages from the ARs in the *AOV* except itself. The number of those ARs is $3i(i+1)$. From each AR, the number of the notification messages is mr_{cc}. The costs of receiving a notification message consist of the cost of receiving the message w_c, and the cost of decryption w_{DE}.

- The Operation Costs of Bogus Requests

The operation costs of an *sending AR* triggered by bogus requests can be expressed as

$$C_{cookie\_f\_sender} = \alpha\beta C_{cookie\_t\_sender} \tag{6.6}$$

α is the percentage of cookies which are eavesdropped and replayed. β, possibility of the replayed cookies being accepted by an AR, can be determined by the probability that the *replay* arrives earlier than the notification.

The time for the arrival of the *replay* is determined by

$$w_{att} + w_w + w_w + w_w \tag{6.7}$$

The attacker first eavesdrops the message with the cost of w_w. Then it parses the cookie and composes a bogus request with the cost of w_{att}. Finally, it sends out the message with the cost w_w. An AR takes w_w to receive the message.

The time for the arrival of a notification message is determined by

$$(w_w + w_{co}) + (w_{EN} + w_c)n + (w_c + w_{DE}) \tag{6.8}$$

For a genuine request, in addition to the cost of $w_w + w_{co}$, the *notification* to the specific AR is the *n*th message, where $1 \leq n \leq 3i(i+1)$. After receiving the *notification* and decrypting it with the total cost of $w_c + w_{DE}$, *notification* is accepted by the AR.

The operation costs of an *receiving AR* which receives *notification* triggered by bogus requests can be expressed as

$$C_{cookie\_f\_receiver} = \gamma\alpha\beta C_{cookie\_t\_receiver} \tag{6.9}$$

γ is the percentage of cells under attack. α is the percentage of requests which are replayed. Note that only the replayed requested which are accepted by an AR can trigger the *notification* messages.

6.2.3 Operation Costs of The AAA Mechanism

The operation costs of genuine requests and bogus requests are discussed separately.

- The Operation Costs of Genuine Requests

  When the cookie mechanism is not applied, the AAAL server is responsible for the authentication. The operation cost of a *sending AR* is

$$C_{aaa\_t\_sender} = mr_{cc}(w_w + w_{EN} + 4w_c + w_{DE}) \tag{6.10}$$

The operation cost of a *receiving AAAL* is calculated as

$$C_{aaa\_t\_receiver} = mr_{cc}(w_{DE} + w_{AUTH} + w_{EN} + 4w_c)(3R(R+1)+1) \quad (6.11)$$

An AAAL server may receive authentication requests from all the cells covered by a MAP domain. From Eq. 6.1, the number of the cells is $3R(R+1)+1$. From each cell, the number of requests is mr_{cc}. For each request, the operation costs consist of the cost of decrypting the message w_{DE}, the cost of authenticating w_{AUTH} and the cost of encrypting the reply message and transmitting the message over the Ethernet interfaces $4w_c$.

- The Operation Costs of Bogus Requests

  The operation costs of an *sending AR* triggered by bogus requests can be expressed as

$$C_{aaa\_f\_sender} = \alpha(mr_{cc}(w_w + w_{EN} + 3w_c)) \quad (6.12)$$

  The operation costs of an *receiving AAAL* triggered by bogus requests can be expressed as

$$C_{aaa\_f\_receiver} = \gamma\alpha(mr_{cc}(w_c + w_{DE} + w_{AUTH})(3R(R+1)+1)) \quad (6.13)$$

  Note that when an authentication fails at AAAL, AAAL discards the requests silently without sending a reply message.

6.2.4 Total Operation Costs

In the cookie mechanism, since an AR may act as both a *sending AR* and a *receiving AR*, the total operation cost of an AR in case that there is no attacking behavior can be expressed as

$$C_{cookie\_t\_total} = C_{cookie\_t\_sender} + C_{cookie\_t\_receiver} \quad (6.14)$$

The total operation cost of an AR in case that there are attacking behaviors can be expressed as

$$C_{cookie\_total} = C_{cookie\_t\_sender} + C_{cookie\_t\_receiver} + C_{cookie\_f\_sender} + C_{cookie\_f\_receiver} \quad (6.15)$$

In the AAA mechanism, an AR acts only as sender of an authentication message whereas the AAAL server is the receiver. Therefore, in case that there is no attacking behavior, the operation costs of an AR and the AAAL server are $C_{aaa\_t\_sender}$ and

Table 6.5: Parameter Values In the Analysis

Parameter	Value	Parameter	Value
w_{EN}, w_{DE}	2	v	1 [m/s]
w_c	1	k	120 m
w_w	5	η	1
$w_{session\_key}$	1	α	1
w_{AUTH}	2	γ	0.2,0.5,1
w_{co}	2	i	1
w_{att}	0,5,10	R	2,3,4

$C_{aaa\_t\_receiver}$ respectively; in case that there are attacking behaviors, the total operation costs of an AR and the AAAL server are

$$C_{aaa\_total\_sender} = C_{aaa\_t\_sender} + C_{aaa\_f\_sender} \tag{6.16}$$

$$C_{aaa\_total\_receiver} = C_{aaa\_t\_receiver} + C_{aaa\_f\_receiver} \tag{6.17}$$

6.2.5 Numerical Results

To calculate the operation costs, the parameters which are defined in Table 6.4 are the the values shown in Table 6.5. The parameters of velocity of a mobile host v and radius of a cell k were referenced from [55, 73].

For the sake of simplicity, constant proportional factor of relation between velocity and cell radius η, percentage of cookies in genuine requests being eavesdropped and replayed α, and the number of ring forming the boundary of *AOV* i are assumed to be 1. Possibility of the replayed cookies being accepted by an AR (i.e. β) can be determined by the time of arrivals of the *notification* and the *replay* shown as Eq. 6.8 and 6.7.

According to the experimental measurements in Chapter 9, time for finding a session key is around $20\mu s$; time for a cookie generation and verification is around $40\mu s$; time for Triple DES encryption/decryption is around $40\mu s$.

Since an authentication with a session key requires a computation of HMAC-SHA1, the authentication time is assumed to equal the time for a cookie verification. When the weight of the session key lookup at AAAL is assumed to 1, the weight of encryption, decryption, cookie verification and authentication with a session key are 2.

The weight of transmitting a 250 bytes over wireless and wired interfaces can be calculated as 5 and 1 respectively.

The number of mobiles in one cell, the weight of attacker's generating a bogus request, parsing the genuine message and composing a bogus message w_{att}, and the percentage of cells under attack γ are variables; the weighted the operation costs at an AR and AAAL are the metrics.

According to Eq. 6.8, the weight of arrival of a *notification* can be expressed as $10 + 3n, 1 \leq n \leq 6$. The time of arrival of a *replay* is weighted as 15, 20 and 25 corresponding to $w_{att} =$ 0, 5 and 10 respectively.

If the weight of a *replay* is smaller than that of a *notification*, the *replay* is assumed to be accepted. Therefore, the probability of the replayed cookies being accepted by an AR are 86$\%$, 53$\%$ and 20$\%$ corresponding to w_{att} = 0, 5 and 10 respectively.

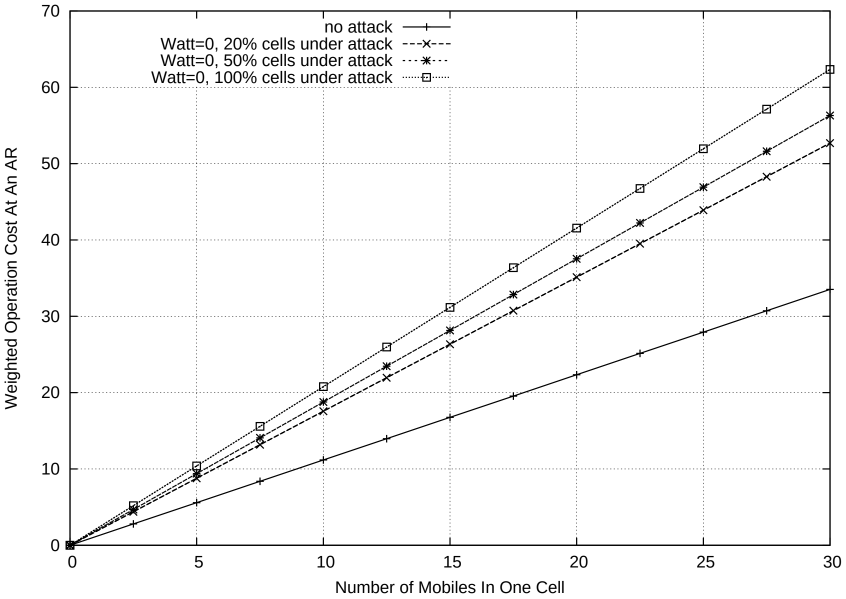


Figure 6.7: Operation Cost at AR in Cookie Scheme: no attack vs. attack w_{att}=0

Figure 6.7 shows the operation costs at an AR in the cookie mechanism when the weighted operation cost at the attacker is fixed. The weighted operation cost of the attack cases is higher than that of the no attack case. The more cells are under attack, the higher is the weighted operation cost.

Figure 6.8 shows the operation costs at an AR in the cookie mechanism when 20% of cell are under attack. The operation cost at an AR when attack is present is always higher than that of no attack cases. Moreover, when an attacker is more powerful, the probability of a replayed cookie being accepted is higher. Therefore, the more powerful attacker results in higher operation cost at an AR.

Figure 6.9 shows the operation costs at the AAAL server in the AAA mechanism when a MAP domain covers two rings of cells. When the number of mobiles in one cell reaches 30, the weighted operation costs at the AAAL are around 300. This value is much greater than that at an AR in the cookie mechanism cases since AAAL is responsible for the all the authentication requests from all the cells in the MAP domain. It is also true that the more cells are under attack, the higher is the operation cost.

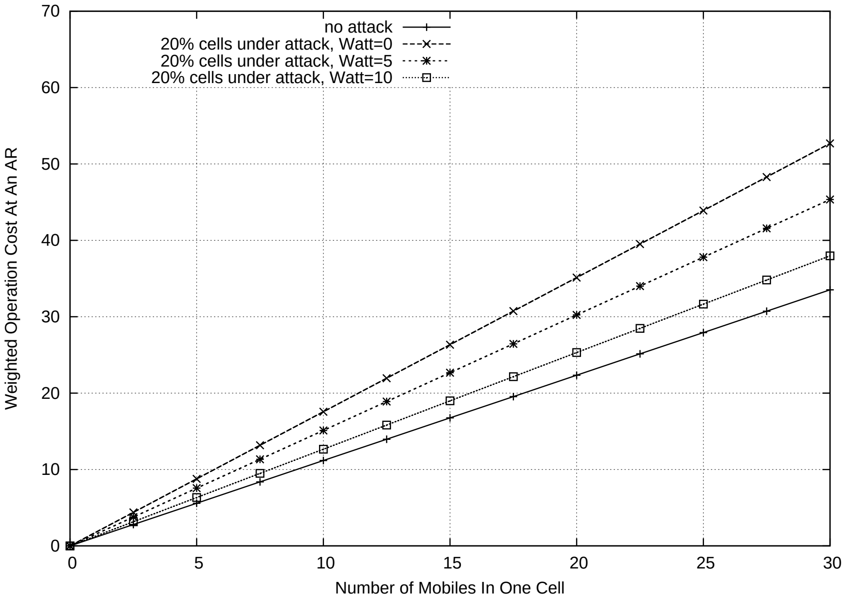


Figure 6.8: Operation Cost at AR in Cookie Scheme: no attack vs. attack γ=20%

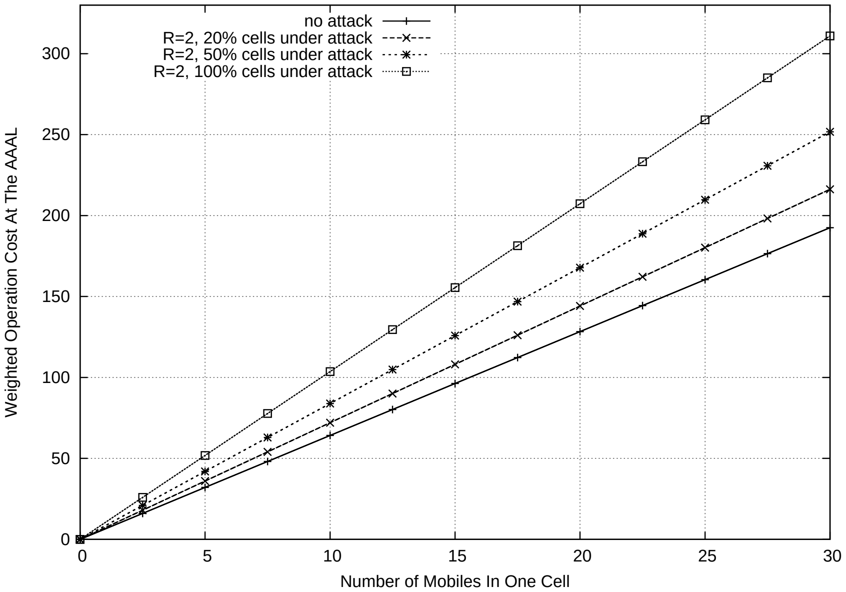


Figure 6.9: Operation Cost at AAAL in AAA Scheme: no attack vs. attack R=2

Figure 6.10 shows the operation costs at the AAAL server in the AAA mechanism when there are 20% cells under attack.It is observed that the more rings a MAP domain covers, the higher is operation cost in both cases of no-attack or attack.

Figure 6.11 shows the operation costs at an AR and the AAAL server when there is no attack to compare the cookie mechanism and the AAA mechanism. Even though the operation costs at an AR in the AAA mechanism is lower than that in the cookie mechanism, operation cost at the AAAL server is very high. The cookie mechanism performs better because the authentication tasks are shared by all the ARs. In the AAA mechanism, however, the tasks are done only at the AAAL server. The more cells in a

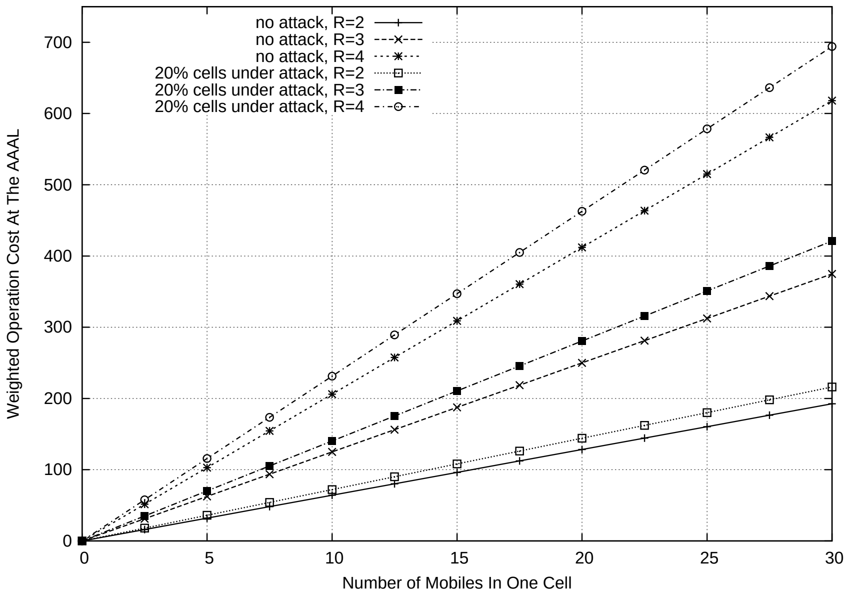


Figure 6.10: Operation Cost at AAAL in AAA Scheme: no attack vs. attack γ=20%

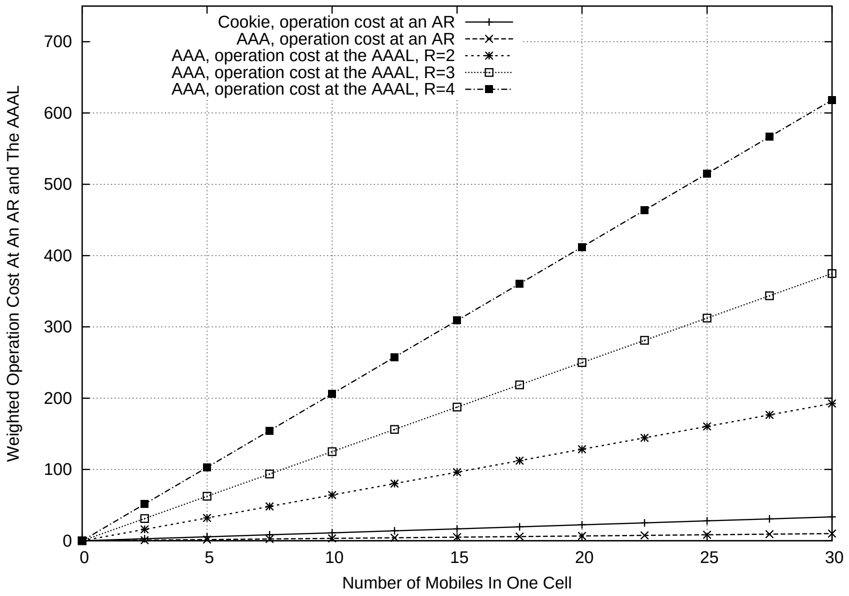


Figure 6.11: Operation Cost at AR and AAAL: no attack

MAP domain, the higher is the operation cost at the AAAL server. The same discussion is applicable in Figure 6.12, which shows the operation costs at an AR and the AAAL server when there are 20% cells under attack.

6.3 Summary

In this chapter, two kinds of Denial of Service (DoS) attacks were identified: DoS due to the resource exhaustion, and DoS due to access control failure.

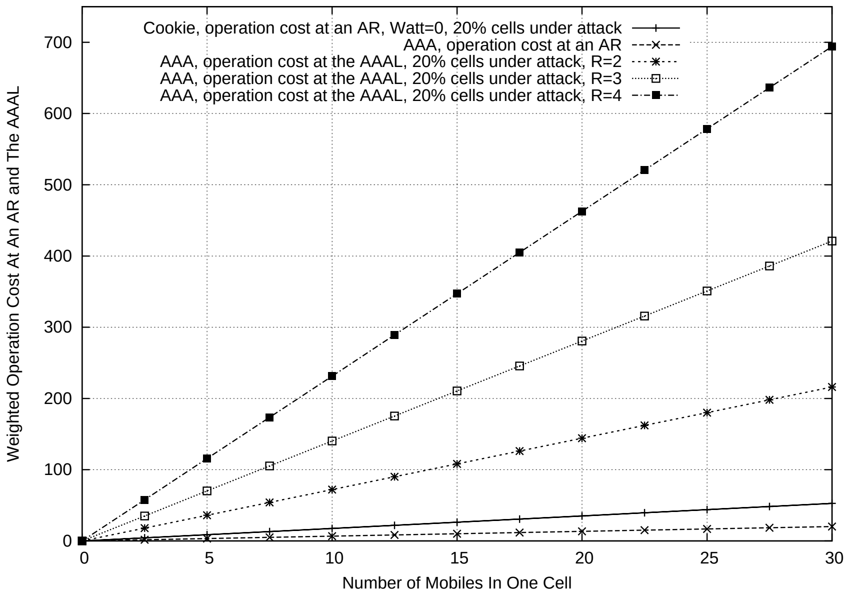


Figure 6.12: Operation Cost at AR and AAAL: attacks are present

A cost-based approach was used to analyze the robustness of the proposed cryptographic protocol against the first DoS attack. $accept_{event}$ and $verification_{event}$ of the protocol were discussed separately in terms of robustness against resource exhaustion DoS. As the result, only MN faces this kind of DoS when it is waiting for the registration answer message in its inter-domain handover procedure. The access network is secure against this kind of DoS in the cryptographic protocol runs.

In dealing with another DoS attack, the operation costs, including the processing cost and the transmission cost were analyzed. A comparison was made between the cookie mechanism and the AAA mechanism. It was proved that the cookie mechanism caused less operation costs when performing authentication than the AAA mechanism.

The probability of a replayed cookie being accepted by an AR essentially depends on how early the *notification* message arrives compared to the *replay*. An effective way to reduce the probability of accepting a replayed cookie (i.e. β) is to hold the operations after a successful cookie verification for a *holding time*. If a notification arrives during the holding time informing that the cookie has been used, the AR knows that the cookie must have replayed to it or other ARs. Therefore, the ARs must hold the re-authorization processes until the second-step authentication finishes. Obviously, the *holding time* should not introduce considerable latency to the whole the re-registration procedure.

This chapter analyze the operation cost at the authentication entities (i.e. AR and AAAL) against DoS due to access control failure. The following chapter will discuss how the security measures performs in the whole re-registration procedure.

Chapter 7

Performance Analysis of the Protocol against Denial of Service

In the preceding chapter, a cost-based approach was used to analyze the cryptographic protocol against Denial of Service (DoS). After the analysis some issues are still open:

- as mentioned in the concluding section of the preceding chapter, the first authentication - verification check of a cookie is not free of cost even though it is a lightweighted preliminary check. An attacker may send bogus messages with wrong cookies and cause an AR to perform such verification checks. When the rate of sending of such bogus messages becomes very intensive, Denial of Service (DoS) may happen;

- how the cryptographic protocol performs in cooperation with other processes (i.e. resource reservation and binding update processes) in the whole registration procedure when DoS attacks exist is unknown, especially when the above-mentioned attacking rate varies;

- in comparison with the two-step authentication approach, how the one-step authentication approach using AAA protocol behaves under such circumstances is unclear.

Therefore, in this chapter queueing theory [49] is used to perform a mathematical analysis of DoS in various re-registration procedures.

The mathematical analysis is carried out based on the network architecture as shown in Figure 7.1 - a Hierarchical Mobile IPv6 (HMIPv6) and Authentication, Authorization, Accounting (AAA) joint architecture.

An Ethernet bus connects an access network and the core network. In the access network, MAP connects ARs via a local bus. AAAL communicates with any AR via MAP. Each AR dominates a cell receiving registration requests from a mobile user via the wireless interface.

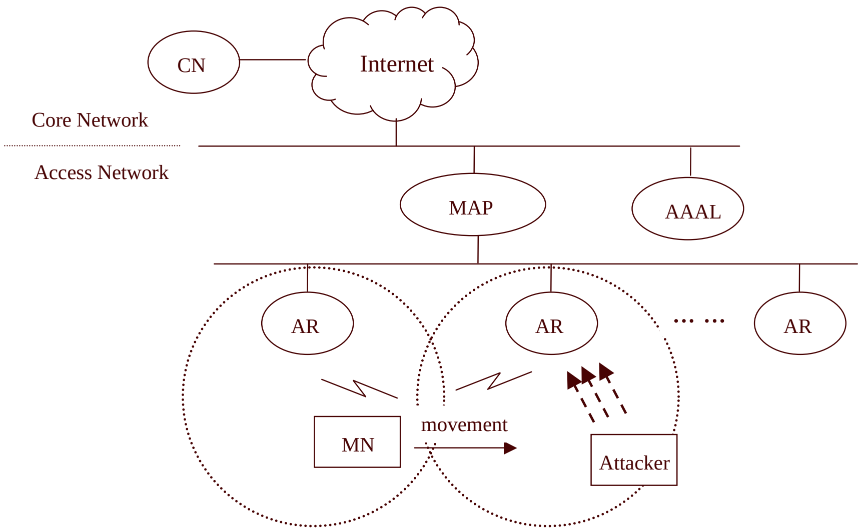


Figure 7.1: An Overview of the Involved Entities

After the MN's first successful registration in the access network, its Authentication and Authorization (AA) information is known to the access network, as well as a session key. Afterwards, it can perform intra-domain handovers, roaming locally from one AR to another.

In the presence of attacking in one cell, an attacker sends bogus request messages to the AR. It is assumed that the bogus request messages contain valid nonces in order to avoid being filled out easily. Thus the AR must pay the *cookiecheck* effort on the bogus requests in the two-step authentication approach (termed as the *cookie protection* case in the following), or perform a strong authentication check at AAAL on the bogus requests in the one-step authentication approach ((termed as the *no cookie protection* case).

7.1 Metrics and Scenarios

To study the performance of a re-registration procedure in terms of registration latency and robustness against DoS, the following three parameters should be examined:

- *Mean response time*: The duration between the transmission of the first bit of a registration request of a legitimate MN and the arrival of the last bit of the corresponding registration response. This parameter can reflect the signaling capacity of a path and the efficiency of an intra-domain handover.

- *Mean queue length at AAAL*: How long in average is the queue of new jobs waiting for service at the AAAL server, in the two-step authentication approach or in the one-step authentication approach (i.e. authentication at AAAL)? This metric indicates that in the one-step authentication approach, there are many conditions under which Denial of Service (DoS) occurs through an overflow of traffic at AAAL, whereas the use of a cookie verification enables AAAL to process only the legitimate requests.

- *Size of the waiting room at AR*: In the one-step authentication approach, when an AR receives a re-registration request, it has to store some data (which is assumed to be 500 bits long) for a request in a *waiting room*, and perform the authentication at AAAL first. When receiving an answer from AAAL, the AR restores the correpsonding data from the *waiting room* and starts the following operations. All the data stored prior to the answered one are removed from the *waiting room* because they are supposed to correspond to bogus requests under the assumption that authentication request should get an answer from AAAL first, unless the authentication check fails and the requests are dropped silently by AAAL. A Markov chain analysis is used to compute the mean value of the total queue length in such a "waiting room".

These parameters are examined in three re-registration procedures as shown in Figure 7.2, Figure 7.3 and Figure 7.4 respectively:

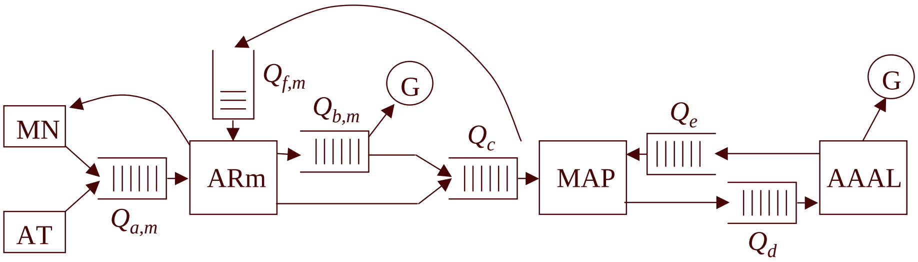


Figure 7.2: Processing Scheme in Case 0

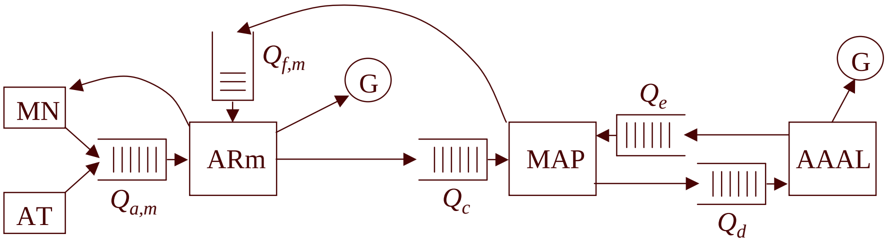


Figure 7.3: Processing Scheme in Case 1

- *Case 0: No cookie protection*:

  In one cell which has one AR, MNs and attackers (AT) send re-registration request messages to the AR (AR_m). The arrival messages are queued in $Q_{a,m}$ of AR_m if

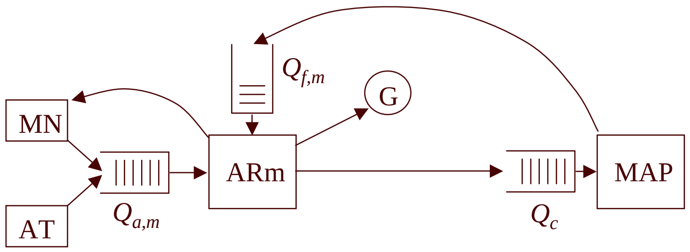


Figure 7.4: Processing Scheme in Case 2

the processor is busy. Since the AA checks need to be performed at AAAL before initiating the source reservation process, AR_m caches temporarily the corresponding QoS request information in $Q_{b,m}$ (which serves as a "waiting room"). When a reply message for a genuine request arrives at AR_m, AR_m discards all the priors in $Q_{b,m}$ since these are regarded as bogus requests, provided all the AA checks at AAAL require the same time. In order to perform Authentication and Authorization (AA) checks at AAAL before processing the request, AR_m generates a new message destined to AAAL and sends the message to Q_c of the MAP. MAP simply forwards the message to Q_d of the AAAL. If the AA checks pass, the AAAL sends an acknowledgement message to Q_e of the MAP. Otherwise, the AAAL drops the request message silently. Obviously, the requests from attackers will be sent to the garbage (denoted by "G"). When MAP receives an acknowledgement message from the AAAL, it forwards it immediately to $Q_{f,m}$ of the AR_m. So far, the security check procedure is complete. Then AR_m starts the joint resource reservation and binding update procedure: it checks its available bandwidth for the QoS request and sends the QoS request to Q_c of the MAP; the MAP performs the same check. It is assumed that AR_m and MAP can also satisfy the QoS request. Thus MAP performs a binding update operation and sends a reply to $Q_{f,m}$ of AR_m. Finally, AR_m sends a registration reply message to the corresponding MN.

- *Case 1: Cookie protection + the joint resource reservation and Binding Update (BU) process and re-authorization process in series*:

  In such a two-step authentication approach, cookie verification serves as the first authentication. After the successful first authentication, the joint resource reservation and Binding Update (BU) process, re-authorization process and the second authentication need to be done. In addition to the second authentication, which will be performed by an AR with the session key, the other two operations may happen in series or in parallel. Case 1 refers to the series case.

  In the AR_m's cell, MNs and Attackers send QoS requests with cookies to $Q_{a,m}$ of AR_m. Before processing the requests, AR_m verifies the cookie. If the verification fails, AR_m drops the request in the garbage ("G") silently. If the verification succeeds, AR_m sends the notification message and starts the resource reservation procedure immediately, checking its available bandwidth and sending the QoS request to Q_c of the MAP. MAP performs the same check and the binding update operation.

Before MAP sends a Binding Acknowledgement (BA) destined to MN, including the QoS related information and the session key, to $Q_{f,m}$ of AR_m, it generates a new message and sends it to Q_d of the AAAL for the authorization check. When the authorization check passes, the AAAL sends a message to Q_e of the MAP, including the session key.

When the MAP receives the message from the AAAL, it then sends a Binding Acknowledgement (BA) to AR_m. When AR_m receives the message from the MAP, AR performs the second authentication check on the re-registration request with the session key. If the check passes, it removes the session key from the reply message. It also generates a new cookie, encrypts it with the session key, inserts the encrypted cookie in the reply message, and forwards it to the corresponding MN.

- *Case 2: Cookie protection + the joint resource reservation, Binding Update (BU) process and re-authorization process in parallel*:

  AR_m performs the same cookie verification when receiving a QoS request from either a MN or an Attacker. If the verification fails, AR_m also drops the request to "G". If the verification passes, the AR_m starts the two processes in parallel. Meanwhile, it sends the notification message.

  When the result of the joint resource reservation and binding update process arrives at AR_m, the result of the re-authorization process may or may not be available at AR_m:

  - if the re-authorization process takes a longer time and no response of the corresponding re-authorization process is available, AR_m sends a reply message to MN setting up the temporary Security Association (SA) to protect the user data, without waiting for the re-authorization result;
  - if the re-authorization process takes a shorter time and the result from the corresponding re-authorization process is already available, AR_m performs the second authentication with the session key, generates a new cookie and encrypts it with the session key. Then AR_m sends a reply message to MN.

Figures 7.5, 7.6 and 7.7 show the process parameters in all three cases. In Case 0 (See Fig. 7.5), shown as Figure 7.5, MN transmits its re-registration request to AR_m

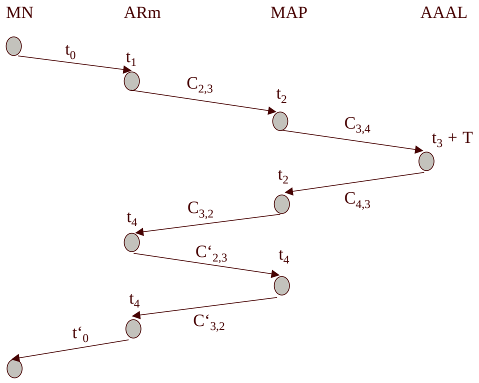


Figure 7.5: Processing Time and Transmission Time in Case 0

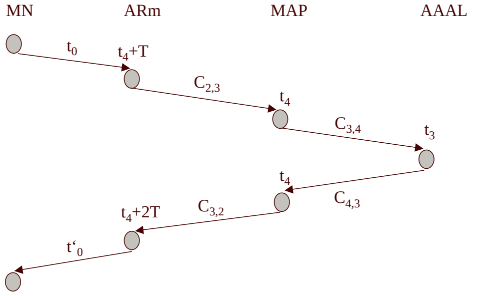


Figure 7.6: Processing Time and Transmission Time in Case 1

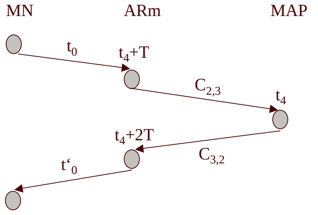


Figure 7.7: Processing Time and Transmission Time in Case 2

over the wireless channel, the transmission time is t_0; then AR_m takes t_1 to generate an Authentication and Authorization (AA) message; the message is transmitted over a 100Mbps link to MAP, the transmission time is denoted as $C_{2,3}$; MAP spends a t_2 time on forwarding the packet towards AAAL; after the transmission time $C_{3,4}$ the packet arrives at AAAL; it takes AAAL (t_3+T) on authorization and authentication; then an answer message is transmitted for a time $C_{4,3}$ and arrives at MAP; MAP spends the same t_2 on forwarding the packet; and after a tranmission time $C_{3,2}$, the packet arrives at AR_m; checking resources and making a reservation takes AR_m or MAP the time of t_4; the transmission times between the two entities are denoted as $C^{'}_{2,3}$ and $C^{'}_{3,2}$ respectively;

Table 7.1: Notations of Processing Time and Transmission Time

Symbol	Remark
t_0	uplink transmission time of the re-registration request over the wireless channel
t_1	time for generating an AA request
$C_{2,3}$	transmission time of the AA request from AR_m to MAP
t_2	time for forwarding an AA request or answer
$C_{3,4}$	transmission time of the AA request from MAP to AAAL
t_3	time for an authorization check
T	time for an authentication check; or generating / verifying a cookie
$C_{4,3}$	transmission time of the AA answer from AAAL to MAP
$C_{3,2}$	transmission time of the AA answer from MAP to AR_m
t_4	time for checking, reserving or confirming resources
$C^{'}_{2,3}$	transmission time of the QoS+mobility packet from AR_m to MAP
$C^{'}_{3,2}$	transmission time of the QoS+mobility packet from MAP to AR_m
$C^{'}_{0}$	downlink transmission time of the re-registration answer from AR_m to MN

AR_m spends t_4 on confirming the reserved resource; the transmission time of the re-registration answer message is $t^{'}_{0}$. The same notations may be applied to Cases 1 and 2, and these notations are summarized in Table 7.1.

In all three cases, λ_{MN} denotes the Poisson rate of message sendings by MNs in a given cell, whereas λ_{AT} denotes the Poisson rate of messages sent by Attackers in a cell that is currently under attack. We also let M denote the total number of cells, whereas N stands for the total number of cells that are currently under attack. The message sending process in a given cell (cell $\sharp\ m$, $1 \leq m \leq M$) is therefore a Poisson process with an intensity λ that is either equal to λ_{MN} (no attack) or to $\lambda_{MN} + \lambda_{AT}$ (attack).

The asymptotic mean arrival rate at $Q_{a,m}$, namely$\lambda^*_{a,m}$, is defined as:

$$\lambda^*_{a,m} = \lim_{t \to +\infty} \frac{\mathbb{E}\left[\sharp(\text{arrivals at } Q_{a,m} \text{ during time}[0,t])\right]}{t},$$

Since the focus of the analysis is the performance of the Layer 3 protocol, the behavior of the MAC protocol of the wireless link, including collisions, is not considered.

It is assumed that the sending of false requests in a cell under attack has an intensity that never exceeds 10% of the transmission capacity of the corresponding wireless channel, so that the effect of collisions along this wireless channel can be neglected.

The asymptotic mean arrival rates at each of the remaining queues (i.e. Q_c, Q_d, Q_e, $Q_{f,m}$) may then be determined as functions of M, N, λ_{MN} and λ_{AT}.

In Case 0:

$$\begin{aligned}\lambda_c^* &= 2M\lambda_{MN} + N\lambda_{AT},\\ \lambda_d^* &= M\lambda_{MN} + N\lambda_{AT},\\ \lambda_e^* &= M\lambda_{MN},\\ \lambda_{f,m}^* &= 2\lambda_{MN};\end{aligned}$$

In Case 1:

$$\begin{aligned}\lambda_c^* &= M\lambda_{MN} = \lambda_d^* = \lambda_e^*,\\ \lambda_{f,m}^* &= \lambda_{MN}\end{aligned}$$

In Case 2:

$$\begin{aligned}\lambda_c^* &= M\lambda_{MN},\\ \lambda_{f,m}^* &= \lambda_{MN}\end{aligned}$$

where M denotes the number of total cells in a MAP domain (geographical zones); N denotes the number of cells under attack; λ_{MN} and λ_{AT} denote the arrival rates of an MN and an attacker respectively.

Since **the mean waiting time of the different queues** are the essential building blocks in the computation of the three metrics (*mean response time*, *mean queue lengths* at AAAL in Cases 0 and 1, and *mean queue length in the "waiting room" of an AR server in Case 0*). For example, the *mean response time* is the sum of waiting time in queues, processing time at nodes adn transmission time over links. While processing time can be taken from exprimental measurement and transmission time can be calculated when packet size and link speed are known, waiting time in queues can be computed by using queueing theory. These mean waiting times will be computed first.

7.2 Computation of Waiting Times

Before coming to the computation of the waiting times at different queues, let us give a presentation of two basic principles from queueing theory.

7.2.1 Introduction of Two Basic Principles

Two basic principles of Queueing Theory shall be used in the computation of waiting times, namely *Litle's theorem* and the *Pollaczek-Khintchine formula*.

Little's Theorem

With the assumption of *ergodicity* (i.e. in the long run, the stochastic mean values of the random variables under consideration resemble their time averages), *Little's Theorem* [10] can be applied in the analysis.

Little's Theorem asserts that, assuming the jobs arrive at a queue Q_i in an ergodic way, the *asymptotic frequency of arrivals* at the Q_i can be defined by

$$\lambda^* = \lim_{t\to+\infty} \frac{\mathbb{E}[\sharp\,(\textit{arrivals during time}[0,t])]}{t}$$

N_t is defined as the number of jobs waiting to be served at time t,

$$N = \lim_{t\to+\infty} \mathbb{E}\left(N_t\right) = \lim_{t\to+\infty} \frac{1}{t}\int_0^t N_u du = \lim_{k\to+\infty} \mathbb{E}\left(N^{(k)}\right)$$

exists, $N^{(k)}$ denoting the number of jobs waiting in Q_i upon the arrival of job $\sharp k$.

Letting $W^{(k)}$ denote the time elapsed between the arrival of job $\sharp k$ in the queue and the beginning of being processed (i.e. the waiting time in the queue), and assuming $\lim\limits_{k\to+\infty} \mathbb{E}\left(W^{(k)}\right) = W$ exists, one has

$$N = \lambda^* \cdot W \tag{7.1}$$

Pollaczek-Khintchine Formula

Since M/G/1 queues can not be cascaded, the higher level nodes (i.e. MAP and AAAL) are not M/G/1 even though the external input to the system is assumed to be Poisson distribution. However, the inputs to the MAP server and the AAAL server are approximated to be Poisson distributed in the analysis based on the following considerations:

- MAP server connects many access routers. Even though one specific flow arriving in Q_c is not Poisson arrival, the overall arrivals can be assumed reasonably to be Poisson according to the **Law of Small Numbers** [80].
- The MAP server forwards packets to the AAAL server in a very short time.
- Since only mean values are used in the analysis, the full richness of the properties of M/G/1 are not considered.

Therefore, the assumption that the mean values of residual service time (RST) and number of jobs in queue (N) seen by an arriving job are equal to those seen by an outside observer [1] can provide a good approximation in the analysis.

Under the assumption, along with *Ergodicity*, the *Pollaczek-Khintchine Formula* [80] can be applied in the analysis. The *Pollaczek-Khintchine Formula* asserts the following: let $W^{(k)}$ denote the waiting time of job $\sharp k$ in Q_i, and $R^{(k)}$ be the time remaining until the server has finished processing the job currently under service (or *residual service time* upon the arrival of job $\sharp k$), and assume $W = \lim\limits_{k\to+\infty} \mathbb{E}\left(W^{(k)}\right)$ and $R = \lim\limits_{k\to+\infty} \mathbb{E}\left(R^{(k)}\right)$ both exist; let also X_k denote the service time of job $\sharp k$ and assume that these variables are identically distributed with finite first moment $\overline{X}$ and second moment $\overline{X^2}$. Then:

[1] this has been proved to be a property of a Poisson arrival process [10]

$$W = \frac{R}{1-\rho}$$

$\rho = \lambda \cdot \overline{X}$ being the *asymptotic utility*. Moreover, R can be computed as the following: (see [10])

$$R = \frac{\lambda^*}{2}\overline{X^2} \tag{7.2}$$

7.2.2 Usage of The Two Basic Principles

Generally, the total time spent waiting in a queue (denoted as Q_i) by job $\sharp k$ when this job just arrived at Node m is determined by the time for the jobs already queueing and waiting for service, together with the time left to complete the job currently under service (i.e. *residual service time* upon the arrival of job $\sharp k$). More precisely, one has

$$\mathbb{E}\left(W_i^{(k)}\right) = \mathbb{E}\left(N_t\right) \cdot \mathbb{E}\left(T_{processing\_time}\right) + \mathbb{E}\left(R_m^{(k)}\right) \tag{7.3}$$

where $\mathbb{E}\left(N_t\right)$ is the mean number of jobs waiting in the queue to be served at time t (i.e. upon arrival of job$\sharp k$); $\mathbb{E}\left(T_{processing\_time}\right)$ is the mean processing time for a job; $\mathbb{E}\left(R_m^{(k)}\right)$ is the mean residual service time at node m upon the arrival of job $\sharp k$ in Q_i.

On the other hand, when k goes to infinity, $\lim\limits_{k\to+\infty} \mathbb{E}\left(W_i^{(k)}\right)$ exists, and this limit will be denoted by $\mathbb{E}\left(W_i\right)$.

According to Equations 7.1 and 7.2,

$$\lim_{t\to+\infty} \mathbb{E}\left(N_t\right) = \lambda^* \cdot \mathbb{E}\left(W_i\right) \tag{7.4}$$

$$\lim_{k\to+\infty} \mathbb{E}\left(R_m^{(k)}\right) = \frac{\lambda^*}{2} \cdot \mathbb{E}\left(T^2_{processing\_time}\right) \tag{7.5}$$

Therefore, according to (7.4) and (7.5), 7.3 asymptotically yields

$$\mathbb{E}\left(W_i^{(k)}\right) = \lambda^* \cdot \mathbb{E}\left(W_i^{(k)}\right) \cdot \mathbb{E}\left(S\right) + \frac{\lambda^*}{2} \cdot \mathbb{E}\left(S^2\right) \tag{7.6}$$

$$\mathbb{E}\left(W_i^{(k)}\right) = \frac{\frac{\lambda^*}{2} \cdot \mathbb{E}\left(S^2\right)}{1 - \lambda^* \cdot \mathbb{E}\left(S\right)} \tag{7.7}$$

where $\mathbb{E}\left(S\right)$ is the mean processing time. Assuming there are two kinds of jobs a and b filling in the queue Q_i, with corresponding asymptotic mean arrival rates denoted by λ_a^* and λ_b^*, and processing times as T_a and T_b respectively; then

$$\mathbb{E}\left(S\right) = \mathbb{E}\left(T_{processing\_time}\right) = \frac{\lambda_a^*}{\lambda_a^* + \lambda_b^*} \cdot T_a + \frac{\lambda_b^*}{\lambda_a^* + \lambda_b^*} \cdot T_b \tag{7.8}$$

The asymptotic mean waiting time at the different queues (i.e. $\mathbb{E}\left(W_{a,m}\right)$, $\mathbb{E}\left(W_{b,m}\right)$, $\mathbb{E}\left(W_c\right)$, $\mathbb{E}\left(W_d\right)$, $\mathbb{E}\left(W_e\right)$ and $\mathbb{E}\left(W_{f,m}\right)$) can be computed using Equations 7.7 and 7.8.

7.2.3 Behavior of The Queue at AAAL

In Case 0:

According to Equations 7.3 and 7.4, the mean waiting time is

$$\begin{aligned}\mathbb{E}\left(W_d\right) &= \mathbb{E}\left(W_d\right)\left\{\left(N\lambda_{AT}\right)T + \left(M\lambda_{MN}\right)\left(T+t_3\right)\right\} \\ &+\mathbb{E}\left(R_{AAAL}\right),\end{aligned}$$

and according to Equation 7.7,

$$\mathbb{E}\left(W_d\right) = \frac{\frac{1}{2}\left\{\left(N\lambda_{AT}\right)T^2 + \left(M\lambda_{MN}\right)\left(T+t_3\right)^2\right\}}{1-\left\{\left(N\lambda_{AT}\right)T + \left(M\lambda_{MN}\right)\left(T+t_3\right)\right\}}. \tag{7.9}$$

In Case 1:

The mean waiting time can be computed in a similar way:

$$\mathbb{E}\left(W_d\right) = \lambda_d^*\mathbb{E}\left(W_d\right)t_3 + \mathbb{E}\left(R_{AAAL}\right),$$

and

$$\mathbb{E}\left(W_d\right) = \frac{\mathbb{E}\left(R_{AAAL}\right)}{1-\lambda_d^* t_3} = \frac{\frac{\lambda_d^*}{2}t_3^2}{1-\lambda_d^* t_3}.$$

7.2.4 Behavior of The Queues at MAP

The asymptotic behavior of the queues at MAP, namely Q_e and Q_c, follows for the three Cases 0,1 and 2.

In Case 0:

From

$$\mathbb{E}\left(W_e\right) = \mathbb{E}\left(R_{MAP}\right),$$

where it is assumed that an arriving message at Q_e can find an empty queue since the processing time at MAP is very small in comparison with that at $AAAL$, the mean waiting times are expressed as the following:

$$\mathbb{E}\left(W_e\right) = M\lambda_{MN}(t_2^2 + \frac{1}{2}t_4^2) + \frac{1}{2}N\lambda_{AT}t_2^2$$

and

$$\mathbb{E}\left(W_c\right) = \frac{\mathbb{E}\left(R_{MAP}\right) + \lambda_e^*\mathbb{E}\left(W_e\right)t_2}{1-(N\lambda_{AT}+M\lambda_{MN})t_2 - M\lambda_{MN}t_4 - \lambda_e^* t_2}.$$

In Case 1:

the mean waiting time at Q_e is

$$\mathbb{E}\left(W_e\right) = \left(\lambda_e^* \mathbb{E}\left(W_e\right)\right) t_4 + \mathbb{E}\left(R_{MAP}\right),$$

which both imply

$$\mathbb{E}\left(W_e\right) = \frac{\mathbb{E}\left(R_{MAP}\right)}{1-\lambda_e^* t_4} = \frac{\frac{\lambda_e^*}{2} t_4^2 + \frac{\lambda_c^*}{2} t_4^2}{1-\lambda_e^* t_4}$$

and

$$\mathbb{E}\left(W_c\right) = \frac{\mathbb{E}\left(R_{MAP}\right) + \lambda_e^* \mathbb{E}\left(W_e\right) t_4}{1-(\lambda_c^+ \lambda_e^*) t_4}.$$

In Case 2:

The mean waiting time is

$$\mathbb{E}\left(W_c\right) = \frac{\mathbb{E}\left(R_{MAP}\right)}{1-\lambda_c^* t_4} = \frac{\frac{\lambda_c^*}{2} t_4^2}{1-\lambda_c^* t_4}.$$

7.2.5 Behavior of the Queues at AR

In Case 0:

The mean waiting times are

$$\mathbb{E}\left(W_{f,m}\right) = \frac{\frac{\lambda_{f,m}^*}{2} t_4^2 + \frac{\lambda_{a,m}^*}{2} t_1^2}{1-\lambda_{f,m}^* t_4}$$

and

$$\mathbb{E}\left(W_{a,m}\right) = \frac{\mathbb{E}\left(R_{ARm}\right) + \lambda_{f,m}^* \mathbb{E}\left(W_{f,m}\right) t_4}{1-\lambda_{a,m}^* t_1 - \lambda_{f,m}^* t_4}.$$

In Case 1:

The situations of a cell under attack and not under attack are different. In a cell that is not under attack, $\lambda_{a,m}^* = \lambda_{MN}$,

the mean residual time is

$$\mathbb{E}\left(R_{ARm}\right) = \frac{\lambda_{MN}}{2}(t_4+T)^2 + \frac{\lambda_{f,m}^*}{2}(t_4+2T)^2,$$

and thus the mean waiting time is computed as

$$\mathbb{E}\left(W_{a,m}\right) = \frac{\mathbb{E}\left(R_{ARm}\right) + \lambda_{f,m}^* \mathbb{E}\left(W_{f,m}\right)(t_4+2T)}{1-\lambda_{MN}(t_4+T) - \lambda_{f,m}^*(t_4+2T)}.$$

On the other hand, in a cell that is currently under attack, $\lambda_{a,m}^* = \lambda_{MN} + \lambda_{AT}$, the mean residual time transforms into

$$\mathbb{E}\left(R_{ARm}\right) = \frac{\lambda_{MN}}{2}(t_4+T)^2 + \frac{\lambda^*_{f,m}}{2}(t_4+2T)^2 + \frac{\lambda_{AT}}{2}T^2,$$

so that finely

$$\mathbb{E}\left(W_{a,m}\right) = \frac{\mathbb{E}\left(R_{ARm}\right) + \lambda^*_{f,m}\mathbb{E}\left(W_{f,m}\right)(t_4+2T)}{1-\lambda_{AT}T-\lambda_{MN}(t_4+T)-\lambda^*_{f,m}(t_4+2T)}.$$

In Case 2:

Similarly to Case 1, for a cell that is not under attack, the mean waiting time is

$$\mathbb{E}\left(W_{a,m}\right) = \frac{\mathbb{E}\left(R_{ARm}\right) + \lambda^*_{f,m}\mathbb{E}\left(W_{f,m}\right)(t_4+2T)}{1-\lambda_{MN}(t_4+T)-\lambda^*_{f,m}(t_4+2T)},$$

whereas in the case of a cell that is currently under attack, the mean waiting time is

$$\mathbb{E}\left(W_{a,m}\right) = \frac{\mathbb{E}\left(R_{ARm}\right) + \lambda^*_{f,m}\mathbb{E}\left(W_{f,m}\right)(t_4+2T)}{1-\lambda_{MN}(t_4+T)-\lambda_{AT}T-\lambda^*_{f,m}(t_4+2T)}.$$

7.3 Computation for Total Response Time

Based on the processing time and transmission time shown in Figures 7.5, 7.6 and 7.7, as well as the waiting time computed in the preceding subsection, the asymptotic mean value of the *total response time* τ can be computed as the following:

In Case 0:

By summing up all the time elements during the re-registration procedure,

$$\begin{aligned}\mathbb{E}\left(\tau\right) = {} & t_0 + \mathbb{E}\left(W_{a,m}\right) + t_1 + C^{(0)}_{2,3} + \mathbb{E}\left(W_c\right) + t_2 + C^{(0)}_{3,4} \\ & + \mathbb{E}\left(W_d\right) + t_3 + T + C^{(0)}_{4,3} + \mathbb{E}\left(W_e\right) + t_2 + C^{(0)}_{3,2} + \mathbb{E}\left(W_{f,m}\right) \\ & + t_4 + C^{('0)}_{2,3} + \mathbb{E}\left(W_c\right) + t_4 + C^{('0)}_{3,2} + \mathbb{E}\left(W_{f,m}\right) + t_4 + t'_0\end{aligned}$$

Based on Table 7.1 $C^{(0)}_{i,j}$ denotes a transmission time in Case 0.

In Case 1:

Similarly,

$$\begin{aligned}\mathbb{E}\left(\tau\right) = {} & t_0 + \mathbb{E}\left(W_{a,m}\right) + t_4 + T + C^{(1)}_{2,3} + \mathbb{E}\left(W_c\right) \\ & + t_4 + C^{(1)}_{3,4} + \mathbb{E}\left(W_d\right) + t_3 + C^{(1)}_{4,3} + \mathbb{E}\left(W_e\right) + t_4 \\ & + C^{(1)}_{3,2} + \mathbb{E}\left(W_{f,m}\right) + t_4 + 2T + t'_0\end{aligned}$$

where $C^{(1)}_{i,j}$ denotes a transmission time in Case 1.

in Case 2:

$$\begin{aligned}\mathbb{E}(\tau) &= t_0 + \mathbb{E}\left(W_{a,m}\right) + t_4 + T + C_{2,3}^{(2)} + \mathbb{E}\left(W_c\right) + t_4 + C_{3,2}^{(2)} \\ &\quad + \mathbb{E}\left(W_{f,m}\right) + t_4 + 2T + t_0'\end{aligned}$$

where $C_{i,j}^{(2)}$ denotes a transmission time in Case 2.

7.4 Computation of Queue Length at AAAL

According to Equation 7.4,

$$\mathbb{E}\left(L_d\right) = \left(N\lambda_{AT} + M\lambda_{MN}\right) \cdot \mathbb{E}\left(W_d\right)$$

Since the mean waiting time $\mathbb{E}\left(W_d\right)$ was already computed as Equation 7.9, the queue length L_d at AAAL in Case 0 can be obtained as

$$\mathbb{E}\left(L_d\right) = \frac{\left(N\lambda_{AT} + M\lambda_{MN}\right)\left\{\left(N\lambda_{AT}\right)T^2 + \left(M\lambda_{MN}\right)\left(T + t_3\right)^2\right\}}{2(1 - \left\{\left(N\lambda_{AT}\right)T + \left(M\lambda_{MN}\right)\left(T + t_3\right)\right)\}}$$

and for Case 1,

$$\mathbb{E}\left(L_d\right) = \frac{\frac{\left(M\lambda_{MN}\right)^2}{2} t_3^2}{1 - M\lambda_{MN} t_3}.$$

7.5 Computation of Queue Length in *Waiting Room*

In the analysis of the length of the "waiting room" $Q_{b,m}$ of Case 0, the only jobs that will be further processed are the "good jobs" corresponding to genuine requests.

On the basis of the scenario given for Case 0, the asymptotic mean total time elapsed between the storage of a “good job” in $Q_{b,m}$ and its marking as a “good job” by the AA acknowledgement message is given by

$$\begin{aligned}\mathbb{E}\left(W_{b,m;good}\right) &= t_1 + C_{2,3}^{(0)} + \mathbb{E}\left(W_c\right) + t_2 + C_{3,4}^{(0)} \\ &\quad + \mathbb{E}\left(W_d\right) + t_3 + T + C_{4,3}^{(0)} + \mathbb{E}\left(W_e\right) + t_2 + C_{3,2}^{(0)} + \mathbb{E}\left(W_{f,m}\right) + t_4\end{aligned}$$

Via Little’s Theorem (see 7.4), the asymptotic mean number of “good jobs” waiting for service in $Q_{b,m}$ is given by

$$\mathbb{E}\left(N_{b,m;good}\right) = \lambda_{b,m;good}^{*}\mathbb{E}\left(W_{b,m;good}\right) = \lambda_{\mathrm{MN}}^{*}\mathbb{E}\left(W_{b,m;good}\right)$$

Taking also the arrivals of “bad jobs” (i.e. fake requests) into account, one may then express the asymptotic total length of $Q_{b,m}$ as

$$\mathbb{E}\left(L_{b,m}\right)=\lambda_{\mathrm{MN}}\mathbb{E}\left(W_{b,m}\right)\left(1+l_{b,m}^{(1)}\right)+l_{b,m}^{(2)},$$

As shown in Figure 7.8, $l_{b,m}^{(2)}$ above denotes the asymptotic mean number of “residual bad jobs” in $Q_{b,m}$ (those “bad jobs” that are located below the lowest "good job" in $Q_{b,m}$), whereas $l_{b,m}^{(1)}$ stands for the asymptotic mean number of "bad jobs" that are located in between two consecutive "good jobs" in $Q_{b,m}$. In Figure 7.8 the number of “residual bad jobs” is 3; when a “good job” is marked through an AA acknowledgement message, the corresponding data is taken away for further treatment, while the jobs prior to it are discarded.

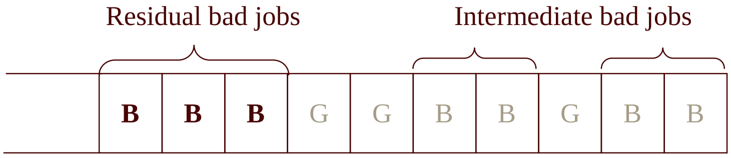


Figure 7.8: Record of Stored Jobs in “Waiting Room”

The mean values $l_{b,m}^{(1)}$ and $l_{b,m}^{(2)}$ can be computed by introducing an appropriate continuous-time Markov chain $\left(\rho_{b,m}^{(t)}\right)_{t\geq 0}$ that is counting the “residual bad jobs” in the course of time. $\left(\rho_{b,m}^{(t)}\right)_{t\geq 0}$ has the transition diagram given in Figure 7.9, and the corresponding transition rates are given by

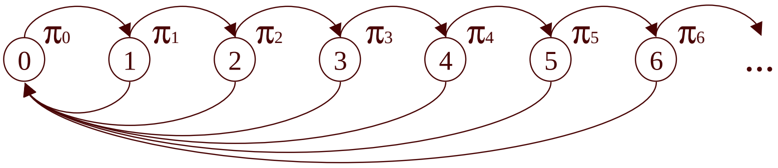


Figure 7.9: The Transission Diagram for $\left(\rho_{b,m}^{(t)}\right)_{t\geq 0}$

$$r_{0,1}=\frac{d}{dt}\mathbb{P}\left(\rho_{b,m}^{(t+dt)}=1|\rho_{b,m}^{(t)}=0\right)=\lambda_{AT}=r_{1,2}=r_{2,3}=\ldots,$$
$$r_{1,0}=\frac{d}{dt}\mathbb{P}\left(\rho_{b,m}^{(t+dt)}=0|\rho_{b,m}^{(t)}=1\right)=\lambda_{MN}=r_{2,0}=r_{3,0}=\ldots$$

Let $\boldsymbol{\pi}=(\pi_0,\pi_1,\pi_2,\ldots)$ denote the invariant probability measure associated with this chain; according to the “global balance equations”:

$$\pi_0 r_{0,1} = \pi_1 r_{1,0} + \pi_2 r_{2,0} + \dots,$$

showing that

$$\pi_0 \lambda_{AT} = (1 - \pi_0)\lambda_{MN}, \quad \pi_0 = \frac{\lambda_{MN}}{\lambda_{MN} + \lambda_{AT}}$$

and

$$\pi_1(r_{1,0} + r_{1,2}) = \pi_0 r_{0,1}, \quad \pi_2(r_{2,0} + r_{2,3}) = \pi_1 r_{1,2}, \dots$$

showing that

$$\pi_k = \frac{\lambda_{MN}}{\lambda_{MN} + \lambda_{AT}} \left(\frac{\lambda_{AT}}{\lambda_{MN} + \lambda_{AT}} \right)^k, \quad \forall k \geq 0.$$

The asymptotic mean number of “residual bad jobs” at $Q_{b,m}$ may thus be computed as

$$l_{b,m}^{(2)} = \lim_{t \to \infty} \mathbb{E}\left(\rho_{b,m}^{(t)}\right) = \sum_{k \geq 1} k\pi_k = \frac{\lambda_{AT}}{\lambda_{MN}}$$

As for $l_{b,m}^{(1)}$, the asymptotic mean number of bad jobs separating two consecutive good jobs in $Q_{b,m}$, it may be seen to satisfy

$$l_{b,m}^{(1)} = \lim_{t \to \infty} \mathbb{E}\left(w_{b,m}^{(t)}\right),$$

where $\left(w_{b,m}^{(t)}\right)_{t \geq 0}$ is an integer-valued process such that

$$\mathbb{P}\left(w_{b,m}^{(t)} \geq k | \rho_{b,m}^{(t)} = l\right) = \begin{cases} \left(\frac{\lambda_{AT}}{\lambda_{AT}+\lambda_{MN}}\right)^{k-l} & k \geq l \geq 0; \\ 1 & l > k \geq 0 \end{cases}$$

Sampling $\left(\rho_{b,m}^{(t)}\right)_{t \geq 0}$ from its equilibrium,

$$l_{b,m}^{(1)} = \lim_{t \to \infty} \sum_{k=1}^{\infty} \mathbb{P}\left(w_{b,m}^{(t)} \geq k\right) = \frac{\lambda_{AT}+\lambda_{MN}}{\lambda_{MN}} - \frac{\lambda_{MN}}{\lambda_{AT}+\lambda_{MN}} + \frac{\lambda_{AT}^2}{(\lambda_{AT}+\lambda_{MN})\lambda_{MN}}.$$

The mean asymptotic queue length at $Q_{b,m}$ is now given by

$$\mathbb{E}\left(L_{b,m}\right) = \left(1 + l_{b,m}^{(1)}\right) \lambda_{MN} \mathbb{E}\left(W_{b,m;good}\right) + l_{b,m}^{(2)}.$$

Table 7.2: Link and Message Length Parameters

Parameter	Value
Wireless link MN↔AR	11/54Mbps
Wired link AR↔MAP↔AAAL	100 Mbps
Wireless PHY+MAC Header	58 bytes
Wired PHY+MAC Header	26 bytes
IPv6 Header	40 bytes
QoS Hop-By-Hop Option	82 bytes
Home Address Option	18 bytes
BU / BU ACK	6 bytes
ESP Header	8 bytes
ESP Authentication Extension	16 bytes
Authenticator	20 bytes
Cookie	32 bytes (HMAC-SHA)

7.6 Results and Interpretation

In this section the metrics derived in the previous section with actual parameters will be evaluated. Table 7.2 shows the assumed link speeds and the message length parameters. Based on the involved signaling protocols and the parameters given in Table 7.2, the message lengths can be computed, and these lengths are used to calculate the transmission times as shown in Table 7.3. Table 7.3 also lists the processing times of individual protocol steps which were obtained by measurements within a prototypical implementation.

Three variables are used in the analysis:

- the total number of cells in a MAP domain,
- the number of cells under attack, and
- the data rate of the wireless link;

Note that the data rates of the wireless link is assumed to be the typical rates of the IEEE802.11 technology.

7.6.1 Total Response Time

Figure 7.10 shows the total response time in relation to the attacking rate per cell when the wireless links are assumed optimistically to be 54 Mbps (physical layer according to the peak data rate of IEEE802.11a, leading to $t_0 \approx 50\mu s$), $M = 50$ cells are connected

Table 7.3: Processing Time and Transmission Time in the Analysis

Symbol		Processing Times and Transmission Times
Case 0	t_0	224 bytes / 11 Mbps or 224 bytes / 54 Mbps
	$C_{2,3}$	256 bytes / 100 Mbps
	$C_{3,4}$	256 bytes / 100 Mbps
	$C_{4,3}$	360 bytes / 100 Mbps
	$C_{3,2}$	360 bytes / 100 Mbps
	$C'_{2,3}$	172 bytes / 100 Mbps
	$C'_{3,2}$	154 bytes / 100 Mbps
	t'_0	186 bytes / 11 Mbps or 186 bytes / 54 Mbps
Case 1	t_0	256 bytes / 11 Mbps or 256 bytes / 54 Mbps
	$C_{2,3}$	204 bytes / 100 Mbps
	$C_{3,4}$	256 bytes / 100 Mbps
	$C_{4,3}$	360 bytes / 100 Mbps
	$C_{3,2}$	186 bytes / 100 Mbps
	t'_0	218 bytes / 11 Mbps or 218 bytes / 54 Mbps
Case 2	t_0	256 bytes / 11 Mbps or 256 bytes / 54 Mbps
	$C_{2,3}$	256 bytes / 100 Mbps
	$C_{3,2}$	360 bytes / 100 Mbps
	t'_0	218 bytes / 11 Mbps or 218 bytes / 54 Mbps
t1		152 μ s *
t2		20 μ s *
t3		152 μ s *
t4		220 μ s *
T		40 μ s *

* The processing time values are obtained from experimental measurements 9.4.1.

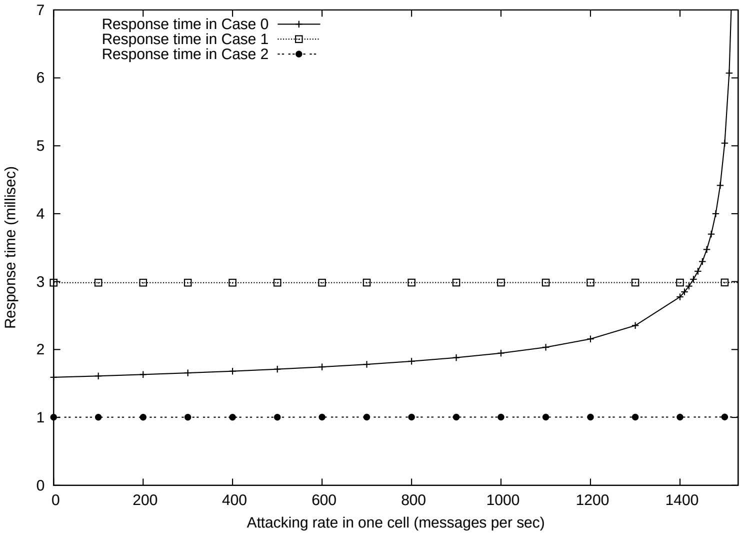


Figure 7.10: Total Response Time: high speed uplink with $t_0 \approx 50\mu s$, M = 50 cells and N = 10 cells under attack for MN rate $x = 40$[messages$/sec$]

to one AAA-server, 10 cells are under attack and $x = 40$[messages$/sec$] are sent by genuine clients. As can be seen, the a DoS situation occurs in case 0 but not in the cookie mechanism cases. In case 0 the total response time grows exponentially when the attacking rate approaches around 1500messages$/sec$. The cookie mechanism requires around one additional millisecond of processing but the system is able to serve genuine clients up to any attacking rate, provided there is still bandwidth left in the corresponding radio cell. Therefore, our DoS protection scheme offers a significant improvement over the unprotected case. As mentioned earlier, we have to be aware of our simplified modeling of the wireless link when interpreting our results. However, 1500messages$/sec$ are well in the valid range for our analysis as they will use less than 10% of the wireless link capacity in case of a 54 MBit/s wireless channel.

In case of a slower wireless uplink with $t_0 \approx 300\mu s$ (corresponding to an IEEE 802.11b WLAN operating at 11 MBit/s with long physical layer preamble) $t_0 \approx 300\mu s$ (corresponding to the peak data rate of IEEE802.11b operating at 11 Mbps). Figure 7.11 again shows that a DoS situation would not occurr before around 1500messages$/sec$ are sent per cell. However, this attacking rate well exceeds the range in which it can be assumed, that the attacker will be able to send his attacking packets over the wireless channel. Therefore, our DoS protection scheme does not offer a benefit in cases with rather low wireless channel capacity, or otherwise stated, more wireless cells have to be supported with one AAA server until our DoS protection scheme offers a significant improvement.

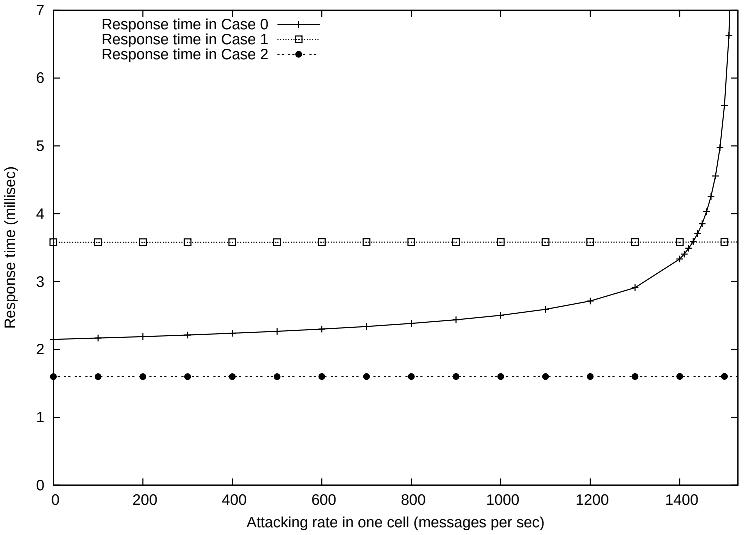


Figure 7.11: Total Response Time: slow speed uplink with $t_0 \approx 300\mu s$, M= 50 cells and N=10 cells under attack for MN rate $x = 40$[messages$/sec$]

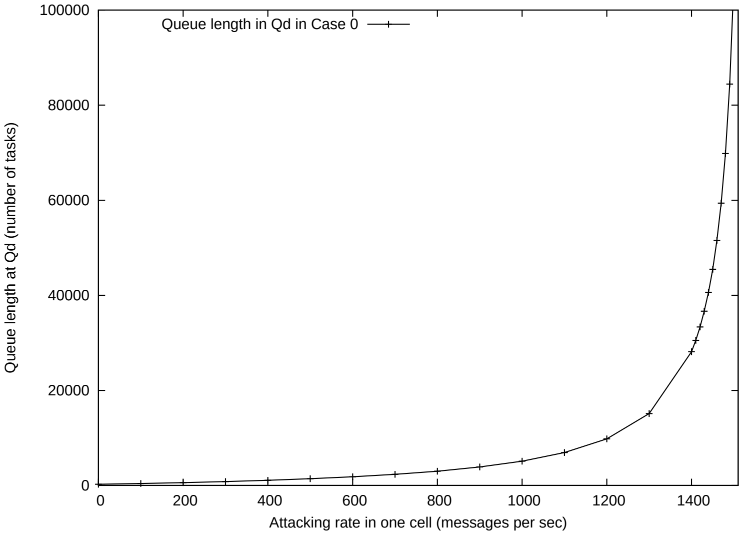


Figure 7.12: Queue Length at AAAL: high speed uplink $t_0 \approx 50\mu s$, M=50 cells and N=10 cells under attack for MN rate $x = 40$[messages$/sec$]

7.6.2 Queue length at AAAL

Figure 7.12 shows the queue length at the AAA server for Case 0 under the same conditions as in Figure 7.10. This graph clearly shows that the DoS situation is caused by the overloading of the AAA server and not by exceeding the transmission capacities of the access network. Under these conditions the AAA server is not able to keep up with checking and discarding bogus messages being sent by attackers, so that the genuine requests from honest clients can not be processed in time.

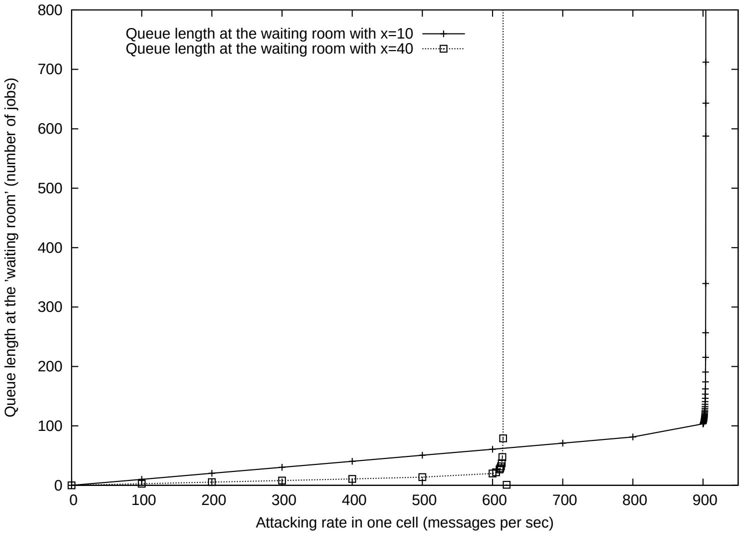


Figure 7.13: Queue Length in *Waiting Room*: high speed uplink $t_0 \approx 50\mu s$, M=50 cells and N=25 cells under attack and different MN rates x

7.6.3 Queue Length in *Waiting Room*

Figure 7.13 shows the queue length in the "waiting room" of an access router in a cell which is under attack (please note that here 25 out of 50 cells are assumed to be under attack). Depending on the rate of genuine requests by honest clients ($x = 10$messages$/sec$ vs. $x = 40$messages$/sec$) a DoS situation will occur earlier. The reason for this behavior lies in our strategy to silently discard bogus requests directly after they have been identified at the AAA server in order to save AAA processing capabilities. This implies, that access routers need to receive a response to a genuine request in order to be able to discard all bogus messages betweeen two genuine requests. Furthermore, it can be seen from this graph that memory depletion situations can occur at access routers under attack. However, the most critical system in the access network infrastructure is the central AAA server.

7.7 Summary

In this chapter, queueing theory including *Little's Theorem* and the *Pollaczek-Khintchine Formula* was applied to analyze the performance of the cryptographic protocol against Denial of Service (DoS). The queue length and the size of "waiting room" were two metrics to indicate the occurence of Denial of Service (DoS) when the sending rate of bogus messages becomes very intensive.

The security messures were integrated in the re-registration signaling protocol. The performance of the protocol was evaluated with the presence of Denial of Service (DoS) attacks. In comparison with another re-registration approach which has a one-step authentication, one may conclude that the cookie protection as a first authentication step in a two-step authentication approach is

- a method to protect against DoS attacks in the QoS reservation process in a distributed scenario;
- a method to speed up registration in the intra-domain handover case by parallelizing QoS reservation process and AA process without introducing additional DoS risks;

Furthermore, our scheme reduces the risk of replayed cookies by implementing an *"area of validity"* in which a cookie is acceptable, and by communicating cookies that have been used once at a particular AR to other ARs in the same area of validity.

The mechanisms of this solution can additionally protect against the following depletion threats

- against depletion of the memory of access routers that would have to maintain state while the authentication of the MN is fetched from the AAAL;
- against depletion of signaling capacity in the access network (by preventing signaling traffic for bogus requests which have not been verified before as being "credible"), and
- against depletion of the resources of the AAAL (by shielding the AAAL from authentication requests which result from bogus QoS requests).

In addition to the robustness against DoS, efficiency of a re-registration procedure including security checks, QoS and mobility signalings is another goal of the thesis. This chapter also addressed the optimization of the procedure by examing the *mean response time* of a legitimate request. In the next chapter, the performance of the proposed re-registration procedure will be compared with other possible solutions focusing on the efficiency issue.

Chapter 8

Performance Analysis of the Protocol in Session State Re-establishment

After the security analysis of the proposal in the preceding chapter, a performance analysis is presented in this chapter in terms of in terms of Registration Response Time, Interruption Time, Packet Loss, CPU processing Load at certain nodes.

Several possible schemes for the re-registration procedure featuring security checks, QoS provisioning and mobility management are formed first. Then a performance analysis is carried out to compare the possible schemes and the proposal by the thesis.

The topology for the analysis is shown in Figure 8.1.

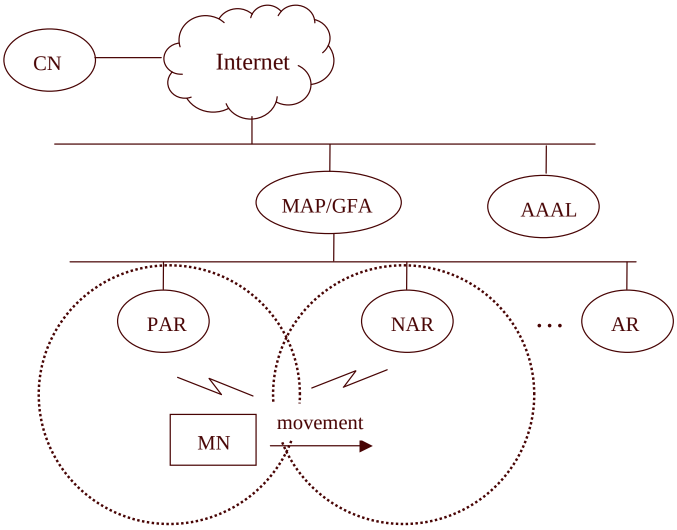


Figure 8.1: The Topology for the Analysis

An Ethernet bus connects an access network and the core network. In the access network, a Gateway Foreign Agent (GFA) connects ARs via a local bus. GFA acts as mobile anchor point (MAP) in the Hierarchical Mobile IPv6 infrastructure. Each AR dominates a cell receiving registration requests from a mobile user via the wireless interface. Local AAA server (AAAL) is the security authority in the access network, responsible for local authentication and authorization (AA). Any AA check request messages from access routers are sent to AAAL via GFA.

MN is assumed to be the receiving node. It receives incoming packets from corresponding node (CN) during a handover. MN is also assumed to be able to compose a stateless IPv6 IP address and send directly a request to either the PAR or the NAR for Duplicate Address Detection (DAD).

8.1 Scheme Description

8.1.1 Guideline to Form the Possible Schemes

In principle, when the access network receives the request, authentication check must be taken on it first. Only the authenticated re-registration requests can be processed further for the mobility management and the session state re-establishment on the new path.

The authentication check can be performed either an local security authority (i.e. the local AAA (authentiation, authorization and accounting) server, termed as AAAL) or an access router (AR) in the access network. In case AAAL performs the authentication, it communicates with an AR by using an AAA protocol; In case an AR performs the authentication, it may get the security information from the old AR by using the Context Transfer (CT) Protocol [57] or it may verify a cookie presented in the request [18].

Mobile IPv4 [75] and Mobile IPv6 [43] were proposed as the main protocols for support IP mobility. To support seamless handovers in IP mobility scenarios, many protocols have been proposed. Fast Handover for Mobile IPv6 (FMIPv6) [51] was designed to minimize handover latency and prevent the degration of QoS. To minimize the mobility signaling cost outside the access network, Hierarchical Mobile IPv6 (HMIPv6) [91] introduced a local entity (i.e. MAP) for the local mobility management. Also it proposed to integrate FMIPv6 with HMIPv6 to minimise the registration latency.

During such a handover, in addition to update the location information of the MN, QoS information needs to be re-established on the new path since an application such as VoIP typically requires some Quality of Service support from the new path (i.e. IntServ), which reserves desirable forwarding treatment to certain distinguished packet streams; or it requires enforcement of a policy at the access router to enable the DiffServ Support.

Therefore, the operations of a re-registration procedure and the correpsonding solutions are summarized in Table 8.1.

Based on Table 8.1, the possible schemes are formed as in six schemes. In case the FMIP is used as the mobility management protocol, either AAA or CT can employed for the authentication check. They also can serve for the authorization check and DiffServ policy deplyment. Therefore, by combining them with a QoS signaling (termed as QoS)

Table 8.1: Potential Solutions for Operations in Re-registration Procedure

Operations	Potential Solutions
Mobility	FMIP, FMIP+HMIP, and HMIP
Security	AAA, CT, and cookie based
Resource reservation	QoS signaling

we have the Schemes 1 and 2: *FMIP+AAA+QoS* and *FMIP+CT+QoS*.

Similarly, in case of "FMIP+HMIP" being used as mobility management protocol, Schemes 3 and 4 are *FMIP+HMIP+AAA+QoS* and *FMIP+HMIP+CT+QoS*. When HMIP is used alone for the mobility management, it can combined to with AAA to form Scheme 5.

The proposal in the thesis which use HMIP alone as mobility management joins the performance comparison as Scheme 6.

One of the possible protocol runs from the Fast Handovers for Mobile IPv6 mechanism [51] is used as the FMIP operation in the analysis. That is, MN is assumed to send first a fast Binding Update(F-BU) message when the protocol starts running. MN is also assumed to be able to receive the fast Binding Acknowledgement (F-BACK) before losing its connection to the previous AR (PAR). The same assumptions are hold for all six schemes.

The details of the six schemes are described in the following subsections.

8.1.2 Scheme 1: FMIP + AAA + QoS

As shown in Figure 8.2, when MN receives advertisements from an AR and notices that the signal strength of the advertisement messages is better than the current one, it decides to switch its link connection. Based on the network information in the advertisements, MN obtains a new CoA (NCoA), while still being connected to the previous AR (PAR). It sends a fast Binding Update (F-BU) message to PAR. PAR may first verify the authenticity of the handover request. Then PAR sends a Handover Initiate (HI) message to NAR. Having verified that the NCoA can be used on the its link, NAR responds with a Handover Acknowledge (HACK) message to PAR. Thus, PAR validates the MN's NCoA and establishes a bidirectional tunnel between the PAR and NAR. While PAR sends a Fast Binding Acknowledgement (F-BACK) message to MN at t_a, it begins forwarding packets intended to MN's previous CoA (PCoA) to the NAR. After MN receives the F-BACK message, it may lose the connectivity with the PAR.

At time of t_b, NAR starts to cache all the forwarded packets until MN establishes connectivity with the NAR at time of t_c. It is assumed that at time t_c, MN establishes an IP connectivity with NAR, and NAR is able to drain the buffer containing the forwarded

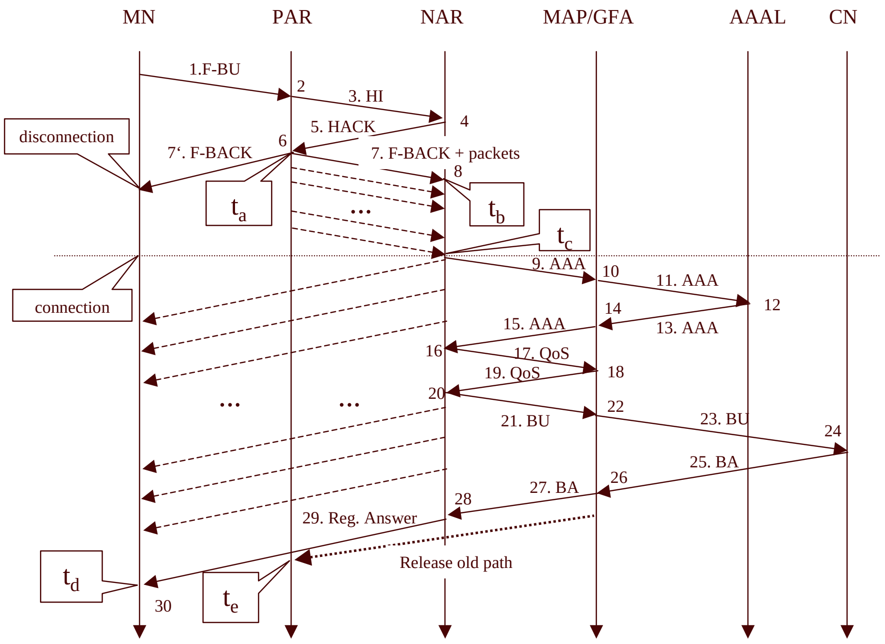


Figure 8.2: Signaling Exchanges of Scheme 1

packets, as well as initiating an AAA process.

Whatever the F-BACK message is received on the old link with PAR or on the new link with NAR, the performance analysis is not afftected. Therefore, MN is assumed to be able to receive the message before losing connection with PAR in the analysis.

Since MN may already transmit its registration request in the F-BU message to PAR, NAR is able to otain the relevant information from PAR and send an AA check message to the local AAA server via GFA. Once the AA checks are successful, the DiffServ information may be deployed from the AAAL to the NAR. Meanwhile, the NAR initiates a QoS resource reservation (e.g. bandwidth) on the path from NAR to the cross-over router, which is the GFA in our case. When the resource is reserved successfully for the session, it performs a binding update process to CN on behalf of MN. When NR gets a successful BA message, it sends a registration answer to MN. At time of t_d, MN receives packets routed directly to its NCoA address. At the time of t_e, PAR receives a message from GFA to tear down the old path.

The essential moments are listed as follows:

- t_a is the time at which PAR starts forwarding packets to NAR;
- t_b is the time at which NAR starts caching the packets forwarded from PAR;
- t_c is the time at which NAR starts draining the buffer containing the forwarded packets to MN;
- t_d is the time at which the handover procedure is complete. Afterwards, packets start coming directly at the new MN's IP address;

- t_e is the time at which PAR stops forwarding packets to NAR;

The duration from t_a to t_c approximates the link switching delay, during which no data packet is sent to MN; the duration from t_b to t_c is then the period during which NAR caches the forwarded packets; the duration from t_c to t_d is the period during which data packets are forwarded to MN by NAR; and the duration from t_a to t_d is the period during which user data have no QoS guarantee and SA protection.

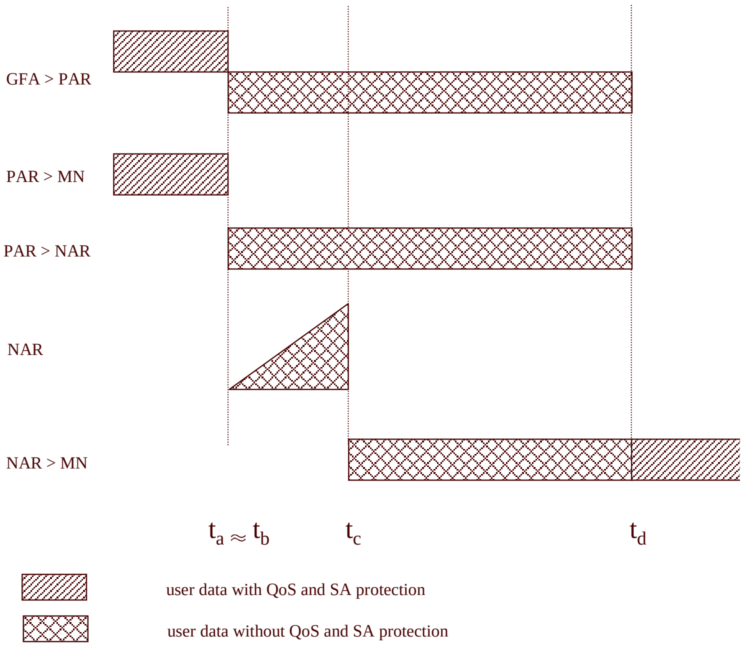

GFA > PAR
PAR > MN
PAR > NAR
NAR
NAR > MN
$t_a \approx t_b$
t_c
t_d
user data with QoS and SA protection
user data without QoS and SA protection

Figure 8.3: User Data Flow During the Registration Procedure in Scheme 1

Figure 8.3 shows the user data flow when MN is the receiving end. It should be noted that, during the period between t_a and t_d, although no packet might be lost, NAR has to consume storage space to cache the forwarded packets before draining the buffer containing the packets to MN, which occurs once MN establishes an IP connectivity with the NAR. Moreover, the forwarded packets from NAR to MN during the peorid between t_a and t_d over the wireless link have no QoS guarantee on the wired QoS path and no protection by a security association between MN and NAR over the wireless link.

Also note that if the security checks or resource reservation fail, the relevant state information needs to be removed and MN has to try with another AR repeating the whole precedure.

The corresponding details are given in Table 8.2.

To simplify the calculation, the packet size of each signaling message is assumed to be 200 bytes except for the first F-BU which is assumed to be 250 bytes. The wireless link is assumed to be 54 Mbps which is the peak data rate of IEEE802.11a. The wired link is assumed to Fast Ethernet link of 100 Mbps.

CN is assumed to be 10 ms away from the access network. The time interval between "disconnection" with PAR and "connection" with NAR (namely link swiching delay and IP connectivity latency) is assumed to be 80 ms according to [52].

Based on the experimental measurement in Chapter 9, the processing time at each node is assumed to be 200 μs except for the forwarding operation which is assumed to be 20 μs for the sake of simplicity.

The same assumptions are held for all the evaluated schemes.

8.1.3 Scheme 2: FMIP + CT + QoS

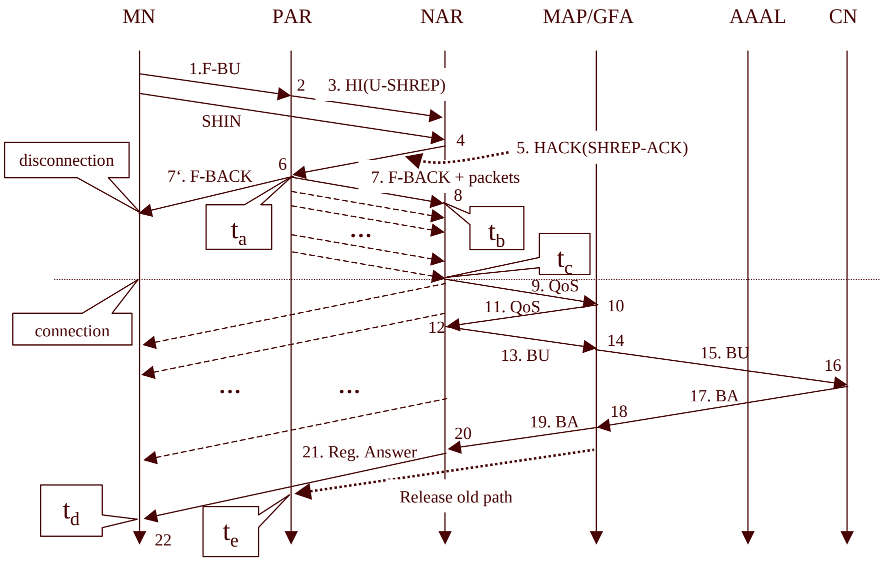


Figure 8.4: Signaling Exchanges of Scheme 2

The combined procedure of FMIP and CT for a seamless handover operation is shown in Figure 8.4. When MN generates its NCoA, it sends a F-BU message to PAR. In this message, MN indicates its desire for context transfer. After PAR has verified the authenticity of the request, it sends a HI message including all the relevant feature contexts as well as the authentication option and an "unsolicited'Seamless Handover Reply" (U-SHREP) option. When MN sends a "Seamless Handover Initiate" (SHIN) message to NAR requesting context transfer, NAR must verify that the authentication data present in the SHIN message matches what was supplied by PAR in U-SHREP. When the check passes, NAR may send a SHREP-ACK option back to PAR in the HACK message. Thus, a bidirectional tunnel between the PAR and NAR is established.

Since it is assumed that there is a security association between PAR and NAR, the authentication and authorization information of the MN has been transferred securely. Hence NAR can perform the security checks without involving the local AAA server. Therefore, the NAR can initiate the resource reservation process directly on the path

Table 8.2: Details of Scheme 1

Step	Name	Time (μs)	Packet Size bytes	Link (Mbps)	Remarks
1	F-BU	-	250	54	F-BU is transferred over a 54 Mbps wireless link
2	PT_{PAR}	200	-	-	PAR takes 200 μs to process the message, including verifying authenticity of the request
3	HI	-	200	100	HI is transferred over a 100 Mbps wired link
4	PT_{NAR}	200	-	-	NAR takes 200 μs to processes the message
5	HACK	-	200	100	HACK is transferred over a 100 Mbps wired link
6 (t_a)	PT_{PAR}	200	-	-	PAR takes 200 μs to process the message
7	F-BACK + packets	-	$\geq$ 200	100	PAR sends F-BACK and forwards packets to NAR via GFA; meanwhile, it sends a F-BACK to MN
8 (t_b)	PT_{NAR}	200	-	-	NAR takes 200 μs to process the message. NAR begins to cache the forwarded packets
9 (t_c)	AAA	-	200	100	NAR sends an AA check message to AAAL via GFA
10	PT_{FGA}	20	-	-	GFA takes 20 μs to forward the message
11	AAA	-	200	100	NAR sends an AA check message to AAAL via GFA
12	PT_{AAAL}	200	-	-	AAAL takes 200 μs to perform the AA check
13	AAA	-	200	100	AAAL sends an AA check response message to NAR via GFA
14	PT_{FGA}	20	-	-	GFA takes 20 μs to forward the message
15	AAA	-	200	100	AAAL sends an AA check response message to NAR via GFA
16	PT_{NAR}	200	-	-	NAR takes 200 μs to process the message. NAR initiates a resource reservation message
17	QoS	-	200	100	NAR sends an resource reservation message to GFA
18	PT_{FGA}	200	-	-	GFA takes 200 μs to process the message
19	QoS	-	200	100	GFA sends an resource reservation response message to NAR
20	PT_{NAR}	200	-	-	NAR takes 200 μs to process the message. NAR initiates a binding update message
21	BU	-	200	100	NAR sends an binding update message to CN
22	PT_{FGA}	20	-	-	GFA takes 20 μs to forward the message
23	BU	10,000	200	100	it is assumed to take 10 ms to transmit BU from GFA to CN
24	PT_{CN}	200	-	-	CN takes 200 μs to perform the BU operation
25	BA	10,000	200	100	it is assumed to take 10 ms to transmit BA from CN to GFA
26	PT_{FGA}	20	-	-	GFA takes 20 μs to forward the message. GFA releases the old path
27	BA	-	200	100	GFA forwards the BA message to NAR
28	PT_{NAR}	200	-	-	NAR takes 200 μs to process the message. NAR sends a registration answer message to MN
29	RegAns	-	200	54	Registration message is transferred over 54 Mbps wireless channel
30 (t_d)	Reception of RegAns	-	-	-	MN receives the Registration message

from NAR to GFA. Afterwards, NAR performs a binding update process to CN on behalf of MN.

t_a, t_b, t_c, t_d and t_e bear the same significance as in Scheme 1.

Details are described in Table 8.3.

8.1.4 Scheme 3: FMIP + HMIP + AAA + QoS

In HMIPv6, a local anchor point (i.e. MAP) is placed in the local network, allowing Mobile IPv6 to benefit from reduced mobility signaling with external networks. The MAP server is then essentially a local home agent, whereas in FMIPv6, the ARs act as local Home Agents which hold binding caches for MNs and receive Binding Updates, so that these ARs function like the MAP specified in HMIPv6. It is also quite possible to have ARs communicating through an aggregation router instead of being directly connected. An aggregation router is then the ideal situation for the MAP functionality [91], so that using MAP in the aggregation router would improve the efficiency of Fast Handovers.

In Scheme 3 where FMIP and HMIP are integrated, it is possible to embed QoS signaling in the binding update process to minimize the registration latency [29].

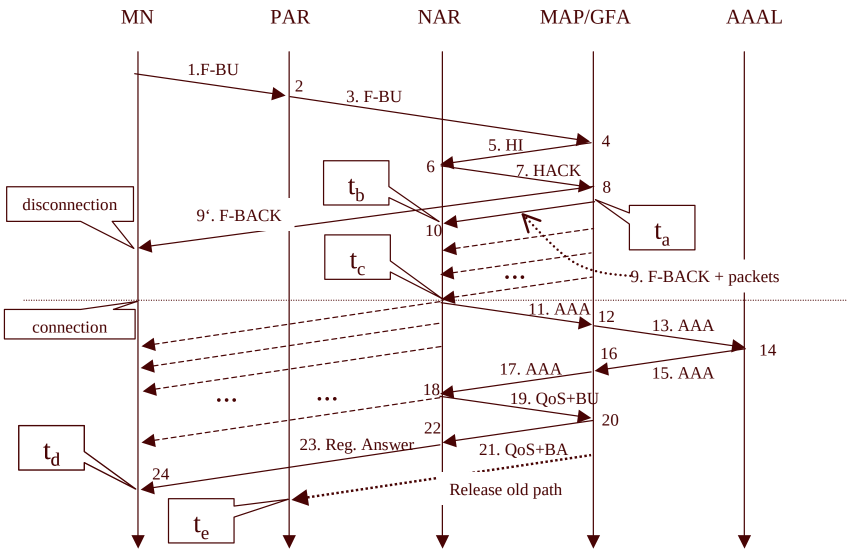


Figure 8.5: Signaling Exchanges of Scheme 3

As shown in Figure 8.5, when MN receives advertisements from NAR and decides to move, it forms its NCoA and sends a F-BU to MAP via PAR. After MAP verifies the authenticity of the handover request, it communicates HI and H-ACK with NAR. Afterwards, it forwards the packets delivered to the old CoA to NAR. Meanwhile it may send a F-Back message to MN via PAR. When the IP connectivity has been set up between MN and NAR, while transferring the cached packets to MN, NAR first

Table 8.3: Details of Scheme 2

Step	Name	Time (μs)	Packet Size bytes	Link (Mbps)	Remarks
1	F-BU	-	250	54	F-BU is transferred over a 54 Mbps wireless link
2	PT_{PAR}	200	-	-	PAR takes 200 μs to process the message, including verifying authenticity of the request
3	HI	-	200	100	HI(U-SHREP) is transferred over a 100 Mbps wired link
4	PT_{NAR}	200	-	-	When receiving SHIN (Smooth Handover Initiate) message from MN, NAR takes 200 μs to perform security checks
5	HACK	-	200	100	HACK(SHREP-ACK) is transferred over a 100 Mbps wired link
6 (t_a)	PT_{PAR}	200	-	-	PAR takes 200 μs to process the message
7	F-BACK + packets	-	$\geq$200	100	PAR sends F-BACK and forwards packets to NAR; meanwhile, it sends a F-BACK to MN
8 (t_b)	PT_{NAR}	200	-	-	NAR takes 200 μs to process the message. NAR begins to cache the forwarded packets
9 (t_c)	QoS	-	200	100	NAR sends an resource reservation message to GFA
10	PT_{FGA}	200	-	-	GFA takes 200 μs to process the message
11	QoS	-	200	100	GFA sends an resource reservation response message to NAR
12	PT_{NAR}	200	-	-	NAR takes 200 μs to process the message. NAR initiates a binding update message
13	BU	-	200	100	NAR sends an binding update message to CN
14	PT_{FGA}	20	-	-	GFA takes 20 μs to forward the message
15	BU	10,000	200	100	it is assumed to take 10 ms to transmit BU from FGA to CN
16	PT_{CN}	200	-	-	CN takes 200 μs to perform the BU operation
17	BA	10,000	200	100	it is assumed to take 10 ms to transmit BA from CN to GFA
18	PT_{FGA}	20	-	-	GFA takes 20 μs to forward the message. GFA releases the old path
19	BA	-	200	100	GFA forwards the BA message to NAR
20	PT_{NAR}	200	-	-	NAR takes 200 μs to process the message. NAR sends a registration answer message to MN
21	RegAns	-	200	54	Registration message is transferred over 54 Mbps wireless channel
22 (t_d)	Reception of RegAns	-	-	-	MN receives the Registration message

intitiates an AAA process to AAAL. When the security check process is successful, it may perform the resource reservation and binding update in one signaling process as described in [29], because the signaling messages of both processes follow the same path from NAR to MAP. When the resource reservation and binding udpate processes have succeeded, NAR sends a registration answer message to MN.

As happens for any FMIP scheme, NAR has to consume storage space during the period of IP connectivity, and the transferred packets from NAR to MN have no QoS guarantee or SA protection.

The time at which MN disconnets from PAR is assumed to be t_a, when MAP fowards packets to NAR instead of PAR. t_b is the time at which NAR starts caching forwarded packets from MAP. NAR forwards the cached packets to MN at t_c, and it initates an AAA process meanwhile. t_d is the endpoint of the handover operation, and the old QoS path is completely torn down at t_e.

Details are described in Table 8.4.

8.1.5 Scheme 4: FMIP + HMIP + CT + QoS

The signaling exchange flow of the combined procedure of FMIP and CT in a Hierarchical Mobile IPv6 architecture is proposed as shown in Figure 8.6.

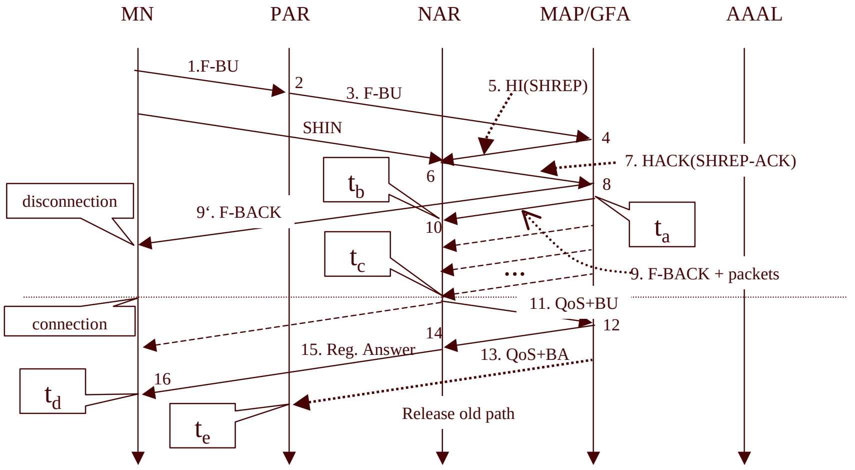


Figure 8.6: Signaling Exchanges of Scheme 4

MN sends a F-BU message to MAP via PAR. Since CT is enabled, after verifying the authenticity of the request, PAR transfers the related context to MAP with the message. MAP then sends a HI(SHREP) message to NAR with the transferred context. Thus when receiving a request from MN, NAR is able to perform authentication and authorzation checks and also deploy the DiffServ Policy. Then if the checks pass, NAR sends a HACK(SHREP-ACK) message to MAP. At time of t_a MAP forwards data packets towarded to MN's old address to NAR. NAR needs to cache the forwarded pack-

Table 8.4: Details of Scheme 3

Step	Name	Time (μs)	Packet Size bytes	Link (Mbps)	Remarks
1	F-BU	-	250	54	F-BU is transferred over a 54 Mbps wireless link
2	PT_{PAR}	20	-	-	PAR takes 20 μs to forward the message
3	F-BU	-	200	100	F-BU is transferred over a 100 Mbps wired link
4	PT_{MAP}	200	-	-	MAP takes 200 μs to process the message
5	HI	-	200	100	HI is transferred over a 100 Mbps wired link
6	PT_{NAR}	200	-	-	NAR takes 200 μs to process the message
7	HACK	-	200	100	HACK is transferred over a 100 Mbps wired link
8	PT_{MAP}	200	-	-	MAP takes 200 μs to process the message
9	F-BACK+packets	-	≥200	100	MAP sends F-BACK and forwards packets to NAR; meanwhile, it sends a F-BACK to MN
10	PT_{NAR}	200	-	-	NAR takes 200 μs to process the message. NAR begins to cache the forwarded packets
11	AAA	-	200	100	NAR sends AA check message to AAAL via MAP
12	PT_{MAP}	20	-	-	MAP takes 20 μs to forward the message
13	AAA	-	200	100	NAR sends AA check message to AAAL via MAP
14	PT_{AAAL}	200	-	-	AAAL takes 200 μs to perform the AA check
15	AAA	-	200	100	AAAL sends an AA check response message to NAR via MAP
16	PT_{MAP}	20	-	-	MAP takes 20 μs to forward the message
17	AAA	-	200	100	AAAL sends an AA check response message to NAR via MAP
18	PT_{NAR}	200	-	-	NAR takes 200 μs to process the message
19	QoS+BU	-	200	100	NAR sends a QoS+BU message to MAP
20	PT_{MAP}	200	-	-	MAP takes 200 μs to process the message; MAP releases the old path
21	QoS+BA	-	200	100	MAP sends a QoS+BU message to NAR
22	PT_{NAR}	200	-	-	NAR takes 200 μs to process the message
23	RegAns	-	200	54	Registration message is transferred over 54 Mbps wireless channel
24 (t_d)	Reception of RegAns	-	-	-	MN receives the Registration message

Table 8.5: Details of Scheme 4

Step	Name	Time (μs)	Packet Size bytes	Link (Mbps)	Remarks
1	F-BU	-	250	54	F-BU is transferred over a 54 Mbps wireless link
2	PT_{PAR}	200	-	-	PAR takes 200 μs to include the related context information and forward the message to MAP
3	F-BU	-	400	100	F-BU is transferred over a 100 Mbps wired link
4	PT_{MAP}	200	-	-	MAP takes 200 μs to process the message
5	HI	-	200	100	HI(U-SHREP) including the transferred context is transferred over a 100 Mbps wired link
6	PT_{NAR}	200	-	-	When receiving a SHIN message from MN, NAR takes 200 μs to perform security checks
7	HACK	-	200	100	HACK(SHREP-ACK) is transferred over a 100 Mbps wired link
8 (t_a)	PT_{PAR}	200	-	-	PAR takes 200 μs to process the message
9	F-BACK + packets	-	≥200	100	PAR sends F-BACK and forwards packets to NAR; meanwhile, it sends a F-BACK to MN
10 (t_b)	PT_{NAR}	200	-	-	NAR takes 200 μs to process the message. NAR begins to cache the forwarded packets
11	QoS+BU	-	200	100	NAR sends a QoS+BU message to MAP
12	PT_{MAP}	200	-	-	MAP takes 200 μs to process the message; MAP releases the old path
13	QoS+BA	-	200	100	MAP sends a QoS+BU message to NAR
14	PT_{NAR}	200	-	-	NAR takes 200 μs to process the message
15	RegAns	-	200	54	Registration message is transferred over 54 Mbps wireless channel
16 (t_d)	Reception of RegAns	-	-	-	MN receives the Registration message

ets for the period between t_b and t_c until MN establishes connection with NAR. Since security checks have been done, NAR then initiates directly a joint QoS and mobility process. At time of t_d when MN receives the registration answer message, the procedure is complete.

Details are described in Table 8.5.

8.1.6 Scheme 5: HMIP + AAA + QoS

In such a case that FMIP is not being deployed, when MN loses its connection to PAR, it has to re-establish a new path with NAR. when MN receives advertisements from NAR and decides to move, it generates its NCoA based on the network prefix information contained in the advertisements. MN needs the same link switching time as in the FMIP cases to gain IP connectivity, which is assumed to be 80 ms according to [52].

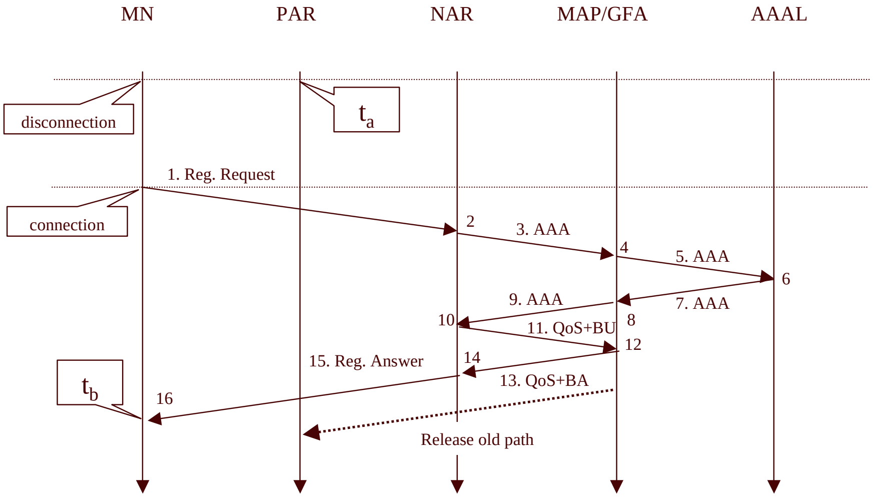


Figure 8.7: Signaling Exchanges of Scheme 5

As shown in Figure 8.7, once MN establishes IP connectivity, it sends a registration request to NAR. NAR first caches some state information such as requested QoS and BU, and initiates an AAA process with AAAL. After a successful security check (i.e. authentication and authorization), while deploying DiffServ policy, NAR may perform resource reservation and binding update in a combined manner. When the process succeeds, NAR sends a registration answer message to MN. The old path is released by MAP after the successful resource reservation and binding update operations.

t_a is the time at which MN loses its connectivity with PAR. After MN gets the registration answer from NAR at t_b, it can receive packets addressed to its new CoA via NAR.

Details are described in Table 8.6.

8.1.7 Scheme 6: HMIP + Cookie + QoS

The case that the re-authorization proceeds slower than “QoS+BU” is defined as Scheme 6.

NAR first performs a cookie verification. A cookie is granted by an access router after MN’s first successful inter-domain handover. The cookie is used to gain access during an intra-domain handover. If the cookie check passes, NAR initiates a combined QoS and BU process and a re-authorization process at the same time. When receiving a positive binding ackowledgement message from MAP, NAR sends a registration answer message to MN, along with a set of SA parameters; this enables setting up a temporary IPSec tunnel without having to wait for the result of the re-authorization process, whenever this result has not yet reached AR.

Table 8.6: Details of Scheme 5

Step	Name	Time (μs)	Packet Size bytes	Link (Mbps)	Remarks
1	Reg. Request	-	250	54	MN sends a registration request over a 54 Mbps wireless link
2	PT_{NAR}	200	-	-	NAR takes 200 μs to process message
3	AAA	-	200	100	NAR sends AA check message to AAAL via MAP
4	PT_{MAP}	20	-	-	MAP takes 20 μs to forward the message
5	AAA	-	200	100	NAR sends AA check message to AAAL via MAP
6	PT_{AAAL}	200	-	-	AAAL takes 200 μs to perform the AA check
7	AAA	-	200	100	AAAL sends an AA check response message to NAR via MAP
8	PT_{MAP}	20	-	-	MAP takes 20 μs to forward the message
9	AAA	-	200	100	AAAL sends an AA check response message to NAR via MAP
10	PT_{NAR}	200	-	-	NAR takes 200 μs to process the message; NAR initiates a QoS+BU message
11	QoS+BU	-	200	100	NAR sends a QoS+BU message to MAP
12	PT_{MAP}	200	-	-	MAP takes 200 μs to process the message; MAP releases the old path
13	QoS+BA	-	200	100	MAP sends a QoS+BU message to NAR
14	PT_{NAR}	200	-	-	NAR takes 200 μs to process the message
15	RegAns	-	200	54	Registration message is transferred over 54 Mbps wireless channel
16 (t_d)	Reception of RegAns	-	-	-	MN receives the Registration message

The handover procedure is regarded to be complete when MN receives the registration message at time t_b.

When the result of the re-authorization process arrives at NAR, NAR performs an authentication check by using the session key. If the security checks pass, NAR generates a new cookie, encrypts it with the session key, and transmits the encrypted cookie to MN, along with a parameter for a new IPSec tunnel, in a registration refresh message. Upon receiving this message at time t_c, MN starts using the new SA to protect the user data and sends a registration refresh reply message to NAR.

The procedure details are illustrated in Figure 8.8.

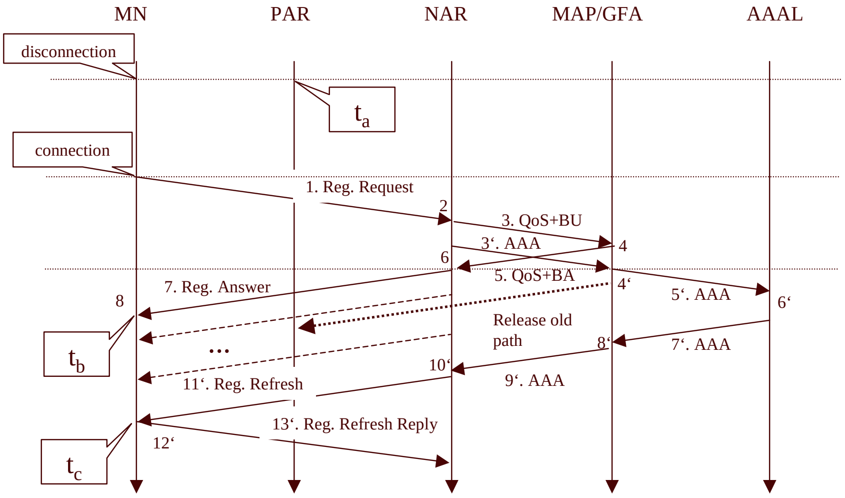


Figure 8.8: Signaling Exchanges of Scheme 6

t_a and t_b denote the same as in Scheme 5. The duration from t_a to t_b is regarded as the registration response time. Additionally, t_b is also the time at which MN starts using the temporary SA with NAR; t_c is the time at which MN starts using the new SA with NAR.

Details are described in Table 8.7.

8.2 Performance Evaluation

The six schemes presented earlier are evaluated using the metrics of Total Response Time, Interruption Time, Packet Loss and CPU Processing Load.

8.2.1 Total Response Time

The *mean value* of the Total Response Time TR, TR denotes the (random) amount of time elapsed between sending of the first bit of a registration request and reception of the last bit of the corresponding registration response. The last bit is received as a period

Table 8.7: Details of Scheme 6

Step	Name	Time (μs)	Packet Size bytes	Link Speed (Mbps)	Remarks
1	Reg. Request	-	250	54	MN sends a registration request over a 54 Mbps wireless link
2	PT_{NAR}	200	-	-	NAR takes 200 μs to perform a cookie check. When cookie verification is successful, it initiates a QoS+BU message and a re-authorization processes in parallel
3	QoS+BU	-	200	100	NAR sends a QoS+BU message to MAP
4	PT_{MAP}	200	-	-	MAP takes 200 μs to process the message; MAP releases the old path
5	QoS+BA	-	200	100	MAP sends a QoS+BU message to NAR
6	PT_{NAR}	200	-	-	NAR takes 200 μs to process the message. It sends a registration answer message including the parameters for the temporary SA
7	Reg. Answer	-	200	54	Registration message is transferred over 54 Mbps wireless channel
8 (t_b)	Receipt of Reg. Answer	-	-	-	MN receives the Registration message
3’	AAA	-	200	100	NAR sends re-authorization check request to AAAL via MAP
4’	PT_{MAP}	20	-	-	MAP takes 20 μs to forward the message
5’	AAA	-	200	100	the request is transmitted over a 100 Mbps wired link
6’	PT_{AAAL}	200 or much more	-	-	AAAL takes 200 μs to perform the re-authorization check check. If the authorization check has to be performed involving MN’s home AAA server, it takes much longer time
7’	AAA	-	200	100	AAAL sends the result of the re-authorization check and the session key to NAR via MAP
8’	PT_{MAP}	20	-	-	MAP spends 20 μs on forwarding the message
9’	AAA	-	200	100	the message is transmitted over a 100 Mbps wired link
10’	PT_{NAR}	200	-	-	MAP takes 200 μs to perform a authentication check with the session key, generate a cookie and encrypt it with the session key. It sends a Registration refresh message to MN
11’	RegRef	-	200	100	Registration Refresh message is transferred over the 54 Mbps wireless channel
12’	PT_{MN}	200	-	-	MN takes 200 μs to process the message
13’	RegRef Reply	-	200	54	Registration Refresh Reply message is transferred over the 54 Mbps wireless channel

endpoint, since MN can receive user data with QoS guarantee and security protection after this moment. Therefore, the TR metric reflects the time needed for complete mobility management and session state re-establishment.

In contrast, as shown in Figure 8.3, although MN may receive user data from NAR much before the whole procedure is complete, all the forwarded data has no QoS guarantee and security protection. Forwarding packets from PAR to NAR enables one to avoid losing packets, which may be crucial to certain packet-loss sensitive applications. thus the time period elapsed between disconnecting and connecting moments will be also examined, defined as the Interruption Time IT in Section 8.2.2

A *homogeneous* fluid flow mobility model is used so that the number of mobile users entering a given cell during one second is a Poisson distributed random variable with mean value x. The homogeneity assumption is valid both in space and time, so that the distribution of this random variable does not depend on the particular cell under consideration, nor on the particular (one second long) time interval.

Looking back at Figure 8.2 (Signaling exchanges in Scheme 1), one may first notice that the variable TR_1 may be expressed as a sum of random variables (queueing times) and deterministic variables (transmission and processing times). t_1 (time needed for the transmission of a 250 bytes long message over the wireless link) can be computed as

$$t_1 = (8 \cdot 250) \cdot (2 \cdot 10^6)^{-1} s,$$

and all the other transmission times may be computed in a similar way, assuming that the wireless channels is able to reach the theoritical capacity of 54 Mbps and the wired channels has a 100 Mbps capacity.

The processing times can be obtained from the assumption shown in Table 8.2. Referring to [52], the time during which NAR holds the packets forwarded by PAR before draining the corresponding buffer has mean value

$$\Delta_1 = 80ms,$$

and that the mean value of the time needed for the transmission of a Binding Update message from MAP to CN and reception of the corresponding answer adds up to $\Delta_2 = t_{23} + t_{24} + t_{25} = 20,2ms$.

It then remains to compute the mean values $E\left[W_{AR}\right]$, $E\left[W_{MAP}\right]$ and $E\left[W_{AAAL}\right]$ of the time that a job needs to spend queueing at an AR server (resp. at MAP, at AAAL) before undergoing service. (According to our homogeneity assumptions, the queueing times at each of the AR servers are identically distributed).

According to the Pollaczek-Khintchine formula (see e.g. Chapter 3 in [10] or Chapter 3 in [49]), the mean residual time at AAAL may be expressed as

$$E\left[R_{AAAL}\right] = \frac{\lambda^*_{AAAL}}{2} \cdot t_{12}^2$$

Here t_{12} still denotes the time required for the execution of task no.12 by AAAL, and

$$\lambda^*_{AAAL} = \lim_{T\to\infty} \frac{1}{T} E\left[\sharp \text{ of job arrivals at AAAL}\right]$$

is the asymptotic mean frequency of arrivals at AAAL. Assuming that the MAP and AAAL servers have been assigned to a region consisting of N cells, one has

$$\lambda^*_{AAAL} = \lambda^*_{MAP} = N \cdot x,$$

whereas $\lambda^*_{AR} = x$.

According to Little's Theorem, the mean number of jobs to be found queueing at AAAL is then given by

$$n_{AAAL} = \lambda^*_{AAAL} \cdot E\left[W_{AAAL}\right],$$

so that the mean value of the time that a job needs to queue at AAAL before being executed is given by

$$E\left[W_{AAAL}\right] = n_{AAAL} \cdot t_{12} + E\left[R_{AAAL}\right] = \frac{(\lambda^*_{AAAL}/2) \cdot t_{12}^2}{1 - \lambda^*_{AAAL} t_{12}}$$

In the very same way, one may use the P-K formula to show that

$$E\left[R_{MAP}\right] = \frac{\lambda^*_{MAP}}{2} \sum_{k=0}^{4} t_{10+4k}^2$$

and then combine it with Little's Theorem in order to obtain:

$$E\left[W_{MAP}\right] = \frac{(\lambda^*_{MAP}/2) \cdot \sum_{k=0}^{4} t_{10+4k}^2}{1 - \lambda^*_{MAP} \sum_{k=0}^{4} t_{10+4k}}$$

Considering anyone of the AR servers in the system, it then remains to compute the mean value of the time a job needs to queue at this server before being treated. Clearly, this AR server is busy treating both jobs corresponding to users entering its cell as well as jobs corresponding to users exiting its cell, and the mean frequency of user exits is just the same as the mean frequency of new entrances, namely x users per second. Proceeding as before,

$$E\left[W_{AR}\right] = \frac{(x/2)\left\{t_2^2 + t_4^2 + t_6^2 + t_8^2 + t_{16}^2 + t_{20}^2 + t_{28}^2\right\}}{1 - x\left\{t_2 + t_4 + t_6 + t_8 + t_{16} + t_{20} + t_{28}\right\}}$$

The mean Total Response Time corresponding to Scheme 1 may finally be expressed as

$$\begin{aligned} E\left[TR_1\right] = \; & 7 \cdot E\left[W_{AR}\right] + 5 \cdot E\left[W_{MAP}\right] + E\left[W_{AAAL}\right] \\ & + \Delta_1 + \textstyle\sum_{i=1}^{29} t_i \end{aligned}$$

The very same methods may then also be used in order to compute the mean values of the Total Response Times $E\left[TR_2\right]$, ..., $E\left[TR_6\right]$ corresponding to all five other Schemes.

8.2.2 Interruption Time

In addition to the Total Response Time, the mean values of the random times during which the User Data flow is being interrupted should also be considered and compared. In the presentation diagrams of Schemes 1, 2, 3 and 4 (see Figs. 8.2, 8.4, 8.5 and 8.6), this Interruption Time (IT) corresponds to the time interval separating t_a and t_c, whereas in the two last Schemes the Interruption Time actually coincides with the Total Response Time. During the Interruption Time, MN is unable to receive any User Data packets.

Thus,

$$IT_5 = TR_5, IT_6 = TR_6,$$

whereas for Schemes 1 and 2:

$$IT_k = t_6 + t_7 + E\left[W_{AR}\right] + \Delta_1$$

for $k = 1, 2$; and for Schemes 3 and 4:

$$IT_k = t_8 + t_9 + E\left[W_{AR}\right] + \Delta_1$$

for $k = 1, 2$.

8.2.3 CPU Processing Load

Considering a server S in the context of Scheme k, the CPU Processing Load $CPU_S^{(k)}$ is defined as the proportion of time during which server S is kept busy processing different jobs in Scheme k. Mathematically speaking, this metric is then simply a linear function of the mean frequency of new arrivals per cell.

For example, the CPU processing load of MAP in the case of Scheme 1 is obtained as:

$$CPU_{MAP}^{(1)} = (Nx) \cdot \sum_{k=0}^{4} t_{10+4k},$$

x still denoting the mean number of new arrivals per second in a given cell, and the service times t_i above being expressed in seconds. The metrics $CPU_{MAP}^{(2)}$, ..., $CPU_{MAP}^{(6)}$ may then be expressed in the very same fashion.

8.2.4 Packet Losses and Storage Space

As shown in Figure 8.2 and Figure 8.3, it is assumed that MN loses its L2 connection with PAR at t_a. At the same time, PAR starts forwarding User Data to NAR. Once a connection is re-established between MN and NAR at t_c, NAR transmits the forwarded packets to MN. Although the forwarded packets have no QoS guarantee and security protection, they will eventually be received by MN. Therefore, there is no packet being lost during the registration procedure in the FMIP schemes (i.e. Schemes 1, 2, 3, and 4), and NAR only needs to store the forwarded packets for the period separating t_a and t_c.

In contrast, when we assume that the handover is performed in a “break before make” manner, MN loses its L2 connection with PAR at t_a as shown in Figure 8.7 and Figure 8.8. It regains the connection with NAR when the registration procedure is complete, i.e. at t_b. In between, i.e. during the Interruption Time elapsed between, the User Data is lost.

Schemes 5 and 6 thus have a specific drawback, namely that of inducing a packet loss during the Re-Registration Procedure. In order to quantify these losses, we have considered the situation where a Mobile User entering a Cell is currently receiving data by using the File Transfer Protocol (FTP) with a rate of $r_{FTP} = 88Kbps$.

The mean packet losses in Schemes 5 and 6 may then be computed simply by multiplying the mean value of the Total Response Time by the appropriate rate:

$$PL_{FTP} = E\left[TR\right] \cdot r_{FTP}$$

In contrast to these losses occuring in Schemes 5 and 6, the requirement of storing temporarily some User Data at MAP in Schemes 1, 2, 3 and 4 may not cause some buffer overflows because in the very worst case of a Region consisting of 50 Cells the buffer space required at MAP for the FTP traffic does not exceed a few Mbits for acceptable values of the traffic intensity variable x ($x \leq 80$ newcomers per second in a cell).

8.3 Numerical Results and Discussion

Considering first the metric of Total Response Time, the three comparative plots are shown in Figure 8.9, Figure 8.10 and Figure 8.11, where the MAP and AAAL servers are being allocated to regions consisting successively of $N = 10$, $N = 30$ and $N = 50$ cells.

In all three cases, it turns out that the HMIP Schemes (i.e. Schemes 3, 4, 5 and 6) have a better TR performance than Schemes 1 and 2 in a regime of moderate traffic intensity, due to the absence of a Binding Update operation involving CN. This shows that hierachical Mobile IP can minimize the response time in micro-mobility senarios.

On the other hand, combining HMIP with FMIP in order to avoid this BU procedure also has the effect of inducing a heavier workload on the MAP server, so that in Schemes 3 and 4 saturation is reached much earlier than in other schemes (Schemes 1, 2, 5 and 6). For example, for $N = 30$ cells per region, $x = 50$ new arrivals per second in a given cell is a saturation threshold in the case of Scheme 3, whereas in Scheme 1, MN may tolerate up to more than 110 new arrivals per second in each cell.

Moverover, as shown in Figure 8.12, in a regime of low traffic intensity (i.e. before the saturation points), the TR values in Schemes 3 and 4 are slightly higher than those

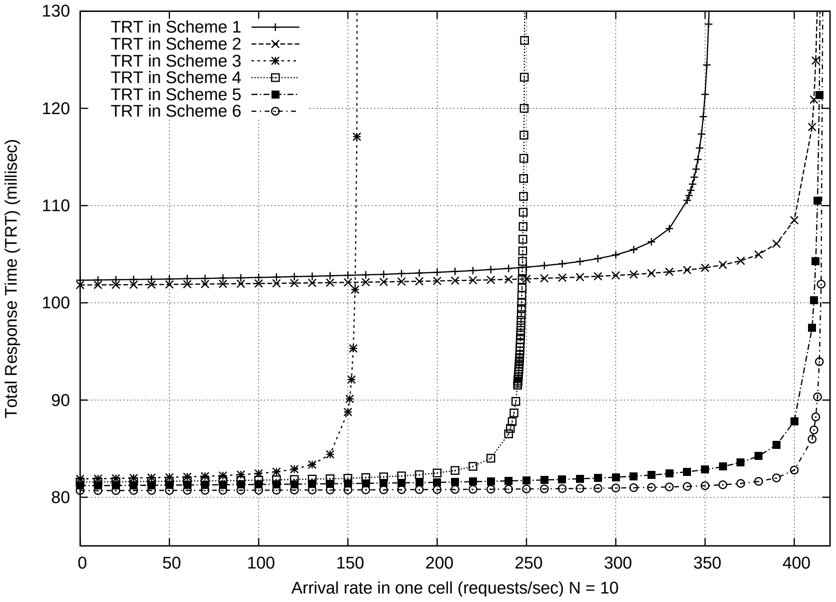


Figure 8.9: Total Response Time N = 10

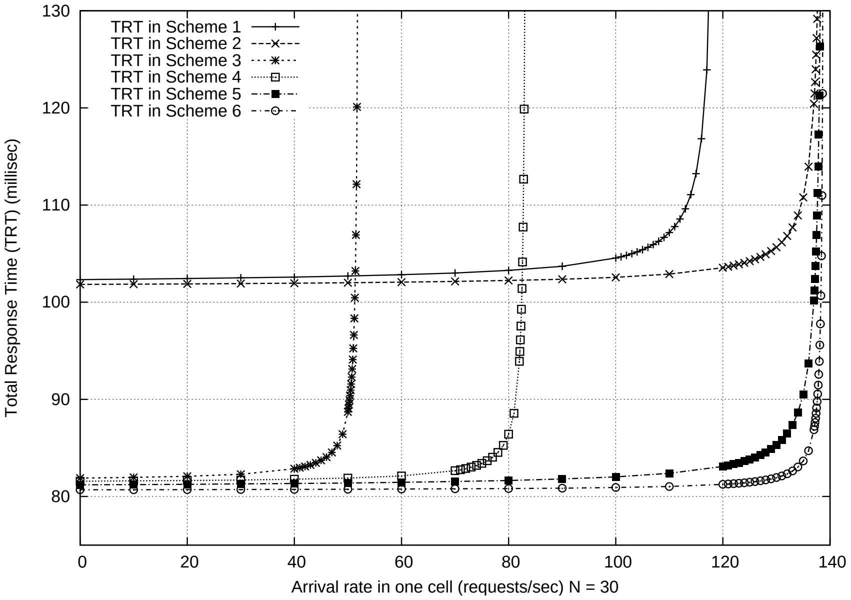


Figure 8.10: Total Response Time N = 30

appearing in Schemes 5 and 6, because MAP has more operations to handle in a fast handover context.

When comparing Schemes 1 and 2, or Schemes 3 and 4, we observe that Context Transfer plays a positive role in reducing the Total Response Time and yielding an improved tolerance to higher traffic intensities.

On the other hand, the TR metric in Scheme 6 (which has a single round trip) is only slightly better than that of Scheme 5 (which has two round trips), since the time for the first round trip in Scheme 5 is very short compared to the link swiching delay.

This difference in the workloads at MAP may also be appreciated through an exami-

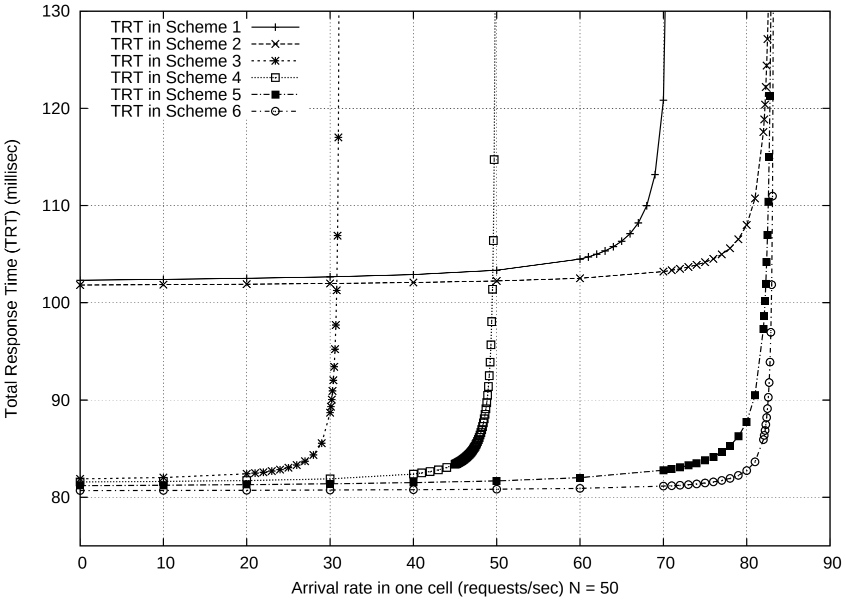


Figure 8.11: Total Response Time N = 50

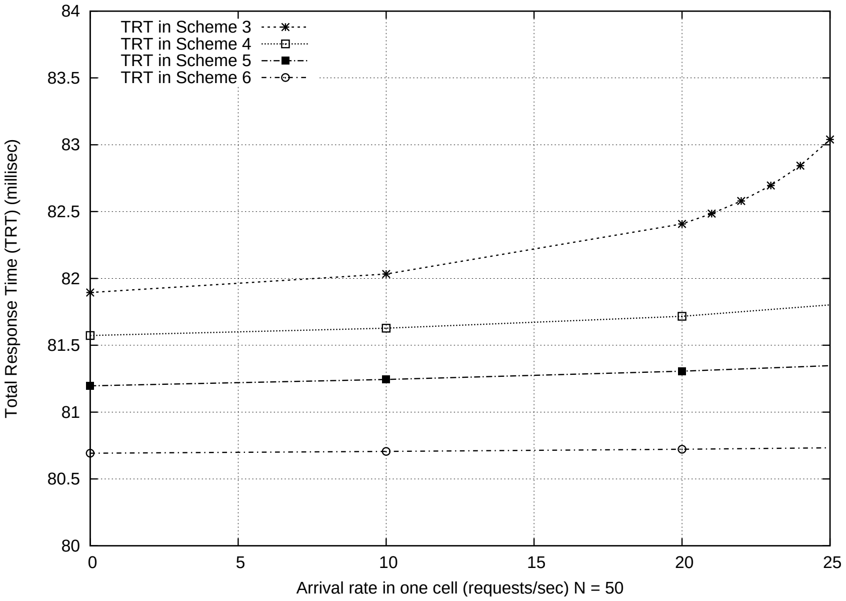


Figure 8.12: Total Response Time in The Regime of Low Traffic Intensity N = 50

nation of the CPU Processing Load at MAP in all six Schemes. As shown in Figure 8.13, Figure 8.14 and Figure 8.15, in all three situations where a region consists of $N = 10$, $N = 30$ or $N = 50$ cells, the GFA/MAP server is more solicited in Scheme 3 and 4 than in other schemes, because of the heavier load induced by FMIP+HMIP operations.

Regarding the Interruption Time metric, as shown in Figure 8.16, the FMIP schemes (i.e. Schemes 1, 2 and 3 and 4) outperform Schemes 5 and 6 because PAR forwards User Data packets to NAR and NAR sends these packets to MN once the Layer 2 connection is established. However, the FMIP schemes have heavier Processing Loads and reach saturation earlier, due to more operations to handle in a Fast Handover context.

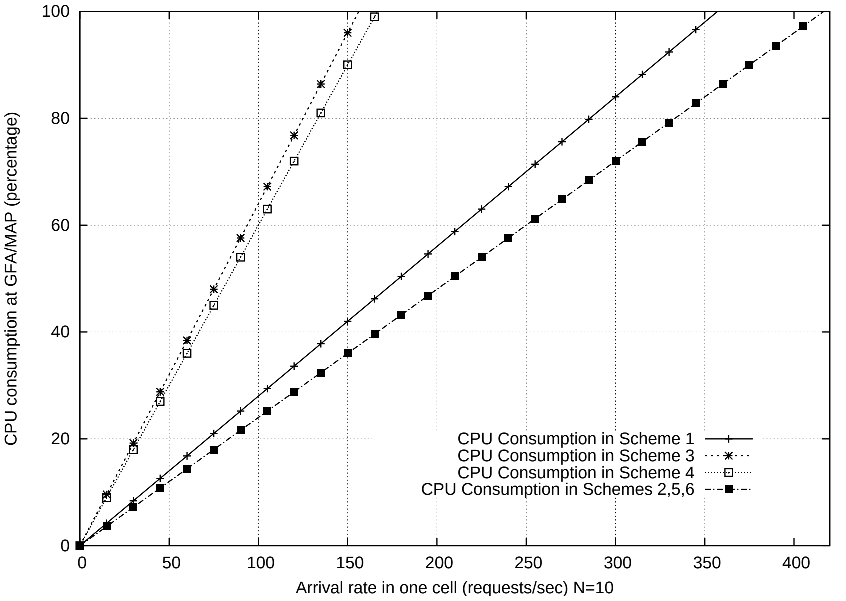


Figure 8.13: CPU Consumption N = 10

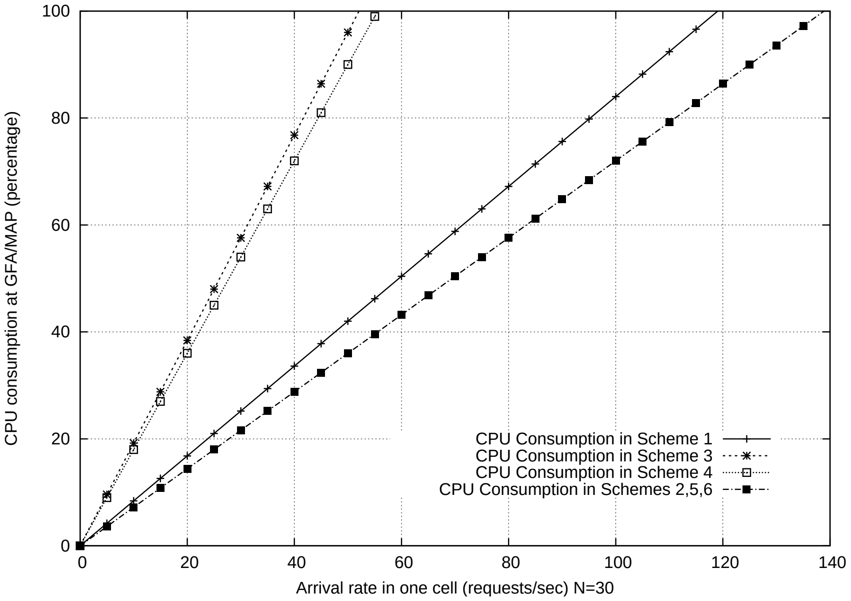


Figure 8.14: CPU Consumption N = 30

Figure 8.17) shows one of the main drawbacks of the HMIP Schemes compared with the FMIP Re-Registration procedures: Schemes 5 and 6 induce a User Data Packet Loss which could be intolerable in certain situations such as File Transfer. Taking fluctuations into account, a *mean value* of $1Kb$ for the packet losses signifies that several Kbits may be lost during a single Re-Registration.

In addition to this analysis of Total Response Time, Interruption Time, CPU Consumption and Packet Loss, we next discuss security aspects and QoS guarantee.

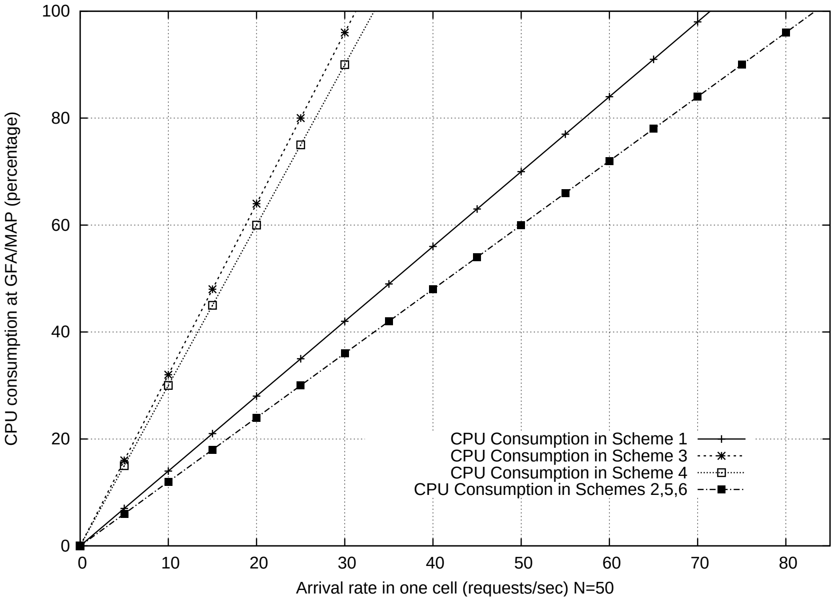


Figure 8.15: CPU Consumption N = 50

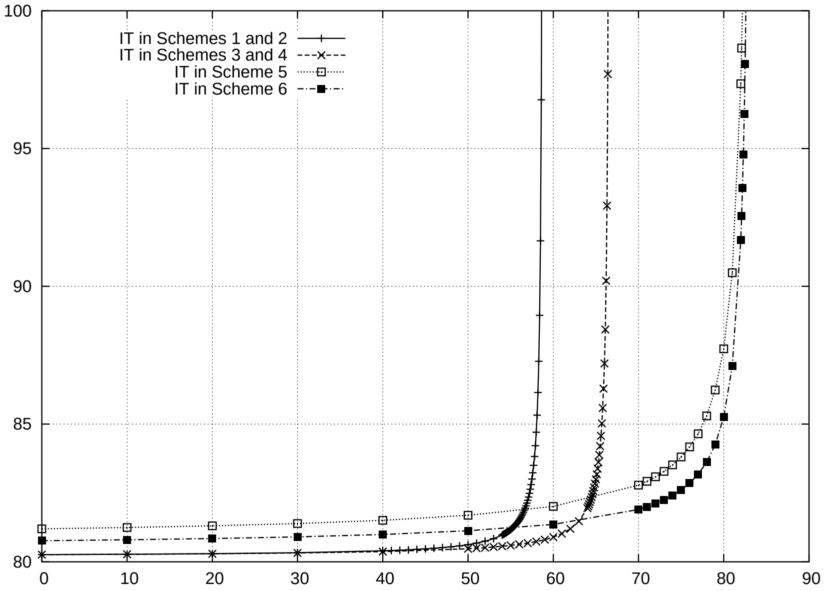


Figure 8.16: Interruption Time N = 50

8.3.1 New Security Association Establishment

In Scheme 1, the new SA between MN and NAR is established only when the registration procedure is complete. At t_c, it starts transmitting the packets being forwarded from PAR without any protection.

In Scheme 2 where CT is applied, the SA between MN and NAR may be established on Step 6 - when security context is transferred from PAR to NAR. However, the SA replies on the session key which are shared between MN and the access network. Both signaling and user data are protected by the same SA.

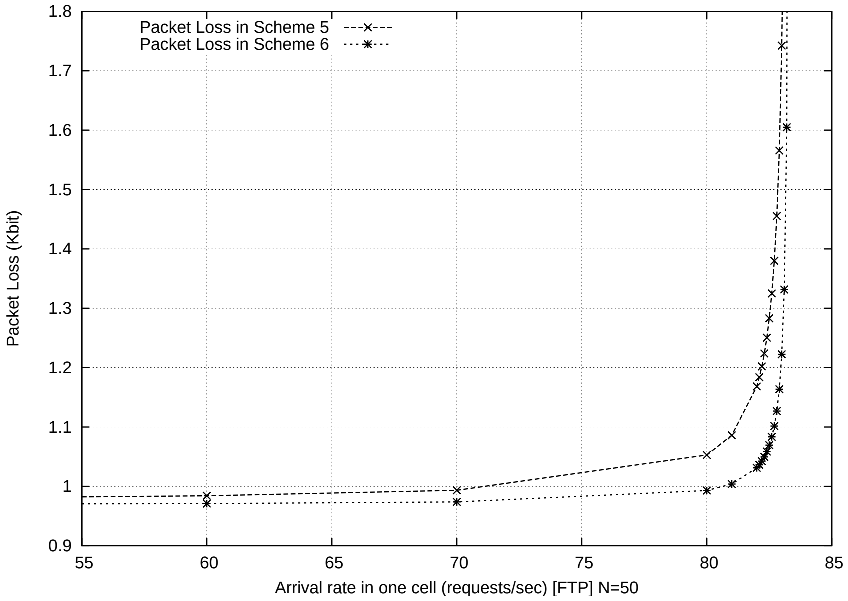


Figure 8.17: Packet Loss of FTP Traffic N = 50

As for the drawbacks of CT itself, when MN upgrades its QoS request on the new path, NAR is unable to authorize the new request so that it has to use an AAA process for this purpose. Moreover, the assumption that any peer of PAR and NAR shares a SA is not always true.

In Schemes 3 and 4, MAP may distribute the session key to NAR to establish the SA between NAR and MN at t_b. Therefore, the packets may be transmitted with protection. However, both signaling and user data reply on the session key being shared between a MN and the access network. Moreover an AAA process has to be undertaken for the new QoS request to be accepted.

In Scheme 5, since MN resumes the connection with the access network only when the registration procedure is complete, the SA between NAR and MN is not established during the period of total response time.

In Scheme 6, although the SA can not be established until the registration procedure is complete, SA may be obtained rather rapidly since MN is able to exchange data securely with the NAR immediately after the QoS path establishement and binding update operations, without having to wait for the completion of the AAA specific security check and establishment of a new definitive SA.

Moreover, in Scheme 6, the signaling and user data are protected with different keys. The key $K_{MN,AN}$ provides integrity/authentication of the uplink as well as confidentiality. The user data are then protected by the temporary key $H(K_{MN,PAR})$ before NAR sends the definitive key $K_{MN,NAR}$ to MN. Hence, even if an attacker somehow comes to know the new temporary key, it can not deduce anything about the definitive key $K_{MN,NAR}$ because the hash function H would have to be inverted. Scheme 5 therefore ensures that potential vulnerabilities arising out of the keys are strictly reduced to the temporary security associations [17].

8.3.2 Robustness Against DoS

When the F-BU request is deemed to be sent to PAR first, as shown in Schemes 1, 2, 3 and 4, the DoS attacks caused by extensive signaling exchanges may be prevented based on the existing SA between a user and its PAR. But whether the risk of DoS attacks in general can be minimized depends on how expensive the authentication check is.

Scheme 5 is facing a serious risk of DoS attacks. Indeed, since an AAA process has to be performed when a request is received by NAR, attackers may trigger extensive AAA processes by sending bogus requests.

To deal with such DoS attacks, a cookie-based mechanism is proposed in Scheme 6 [18].

8.3.3 QoS Guarantee for Delivered Packets

Before registration procedure is complete, in Scheme 1 and 2 where Fast handover scheme is applied, User Data destined to old CoA are transmitted continuously from CN to PAR. Then the packets are forwarded from PAR to NAR via GFA. In Schemes 3 and 4, packets which are coming in continuously are delievered to NAR via MAP. The forwarded packets are bufferred by NAR until L2 connectivity between NAR and MN is set up. Afterwards, the packets are drained to MN.

Two facts influence QoS guarantee for packets.

- *Buffering by NAR:* if the data packets of real-time applications are buffered for a longer time period than the admissible end-to-end delay, they may become useless;

- *Traversing on a non QoS path:* before the new QoS path from GFA to NAR is set up, the packets are delivered without QoS guarantee.

Therefore, it is essential to manage the forwarding buffer efficiently and minimize the registration latency.

In Schemes 5 and 6, assuming handovers are performed in a “break before make” manner, although MN loses some packets during the interruption time it is receiving User Data with QoS guarantee, since all the incoming packets are using the new QoS path.

8.4 Summary

In this chapter, the proposal in the thesis was comparied with the other five Re-Registration Schemes featuring FMIP, Context Transfer and HMIP, using the metrics of Total Response Time (*TR*), Interruption Time (*IT*), CPU Consumption at the MAP/GFA Server and Packet Loss (*PL*). Finally, the SA establishment, robustness against DoS and QoS guarantee on delivered packets were discussed.

It turns out that Schemes 5 and 6 (based on HMIP) provide lower Total Response Times, with a better tolerance to higher values of the Traffic Intensity. These two Schemes may also be seen to offer more robustness when comparing the CPU Processing Loads induced by each of the five Schemes at the MAP/GFA Server. Scheme 5 should be preferred when taking also robustness against DoS attacks into consideration. Moreover, in Scheme 5 the Packet Loss and Interruption Time may be further reduced if MN is able to keep its connection to NAR for a longer time before the new QoS path is set up.

On the other hand, when considering lower values of the Traffic Intensity and restricting our attention to the *IT* metric, the comparison becomes slightly favorable to Schemes 3 and 4, in which FMIP and HMIP are integrated.

Furthermore, one should also realize that in Schemes 5 and 6 several Kbits of User Data may be lost during a single Re-Registration, even when considering intermediate values of the Traffic Intensity and FTP Data only, which might lead one to choose one of the first three FMIP Schemes (which are able to transmit all User Data to MN) for certain Packet Loss sensitive applications.

From the plots of all metrics, we conclude that the bundling feature of context transfer on FMIP can improve the performances. Assigning fewer Cells to each MAP server results in an improved tolerance to higher traffic intensities and less Packet Losses

Conclusively, for the realtime applications which are delay sensitive and packet loss insensitive, the proposal in the thesis (i.e. Scheme 6) is the most preferable solution. Moreover, when the tolerance of work load, security association establishement and QoS guarantee, the proposal in the thesis featuring a cookie exchange procedure and a temporary SA establishment, is also the most preferable in a secure, QoS-aware macro-mobility scenario.

Chapter 9

Experimental Evaluation

9.1 Introduction

The introduction section first describe the goals and challenges of the experimental evaluation. Then the methodology is presented.

9.1.1 Goals and Challenges

The analysis chapters necessitate parameters of different processing times. The parameters should be given by the experimental experiences. Furthermore, the following questions should be asked by a prototypical implementation:

- how differently does the the CASP Mobility Client protocol perform from the QoS-conditionalized Binding Update approach [29] in terms of total response time for a QoS request? It is interesting to compare the two QoS signaling protocol to determine how the general end-to-end approach in QoS signaling design behave in intra-domain handovers occurred in the specific Hierarchical Mobile IPv6 (HMIPv6) infrastructure. In such a handover scenario, the optimization of the QoS-conditionalized Binding Update approach has been proved in [68].

- how does the cookie mechanism behave in the presence of the DoS attacks in terms of the total response time? The cookie verification served as a preliminary authentication check is one of the most important features of the formulated proposal. It is important to evaluate its performance by means of implementation and simulation.

In summary, the goals of the experimental evaluation include

- providing parameters to the theoretical analysis;

- comparing the end-to-end QoS signaling approach with the peer-to-peer approach in intra-domain handovers in an HMIPv6 environment, by evaluating the CASP Mobility Client protocol and the QoS-conditionalized BU approach;

- comparing the behaviors of the different approaches in the presence of DoS attacks.

9.1.2 Methodology

The compared QoS signaling schemes include the *CASP Mobility Client* protocol and *QoS-conditionalized BU* approach, and the security schemes in the evaluation are *AAA* protocol and *cookie* mechanism. All the schemes for a secure and QoS-aware mobility support in intra-domain handovers are evaluated in both cases that DoS attacks exist or do not exist.

The selected studied cases for the experimental evaluation are shown in Table 9.1.

Case No.	QoS and Mobility Signaling	Security Measures	Presence of DoS Attacks
1	QoS-conditionalized BU	cookie	No
2	QoS-conditionalized BU	AAA	No
3	CASP Mobility Client Protocol	cookie	No
4	CASP Mobility Client Protocol	AAA	No
5	QoS-conditionalized BU	cookie	Yes
6	QoS-conditionalized BU	AAA	Yes
7	CASP Mobility Client Protocol	cookie	Yes
8	CASP Mobility Client Protocol	AAA protocol	Yes

Table 9.1: Selected Cases for Experimental Evaluations

Measurement and simulation are used to evaluate the performance of the cookie mechanism. Among the eight cases, two cases (i.e. Cases 5 and 6) are evaluated with a simulation using OMNET++, and others with a prototypical implementation.

An overview of the methodology for the experimental evaluation is given in Figure 9.1. As shown in this figure, the experiment environment include prototypical implementation and simulation. For the prototypical implementation, the selected studies cases 1 and 2 use the *QoCoo* (i.e. QoS-conditionalized BU) environment [68], whereas Cases 3, 4, 7 and 8 use the *CASPMob* (i.e. CASP Mobility Client Protocol) Environment. Both *QoCoo* and *CASPMob* are built on the “Mobile IP for Linux (MIPL) Environment” from the Helsinki University of Technology [48]. For the simulation, Cases 5 and 6 use simulation model in OMNET++.

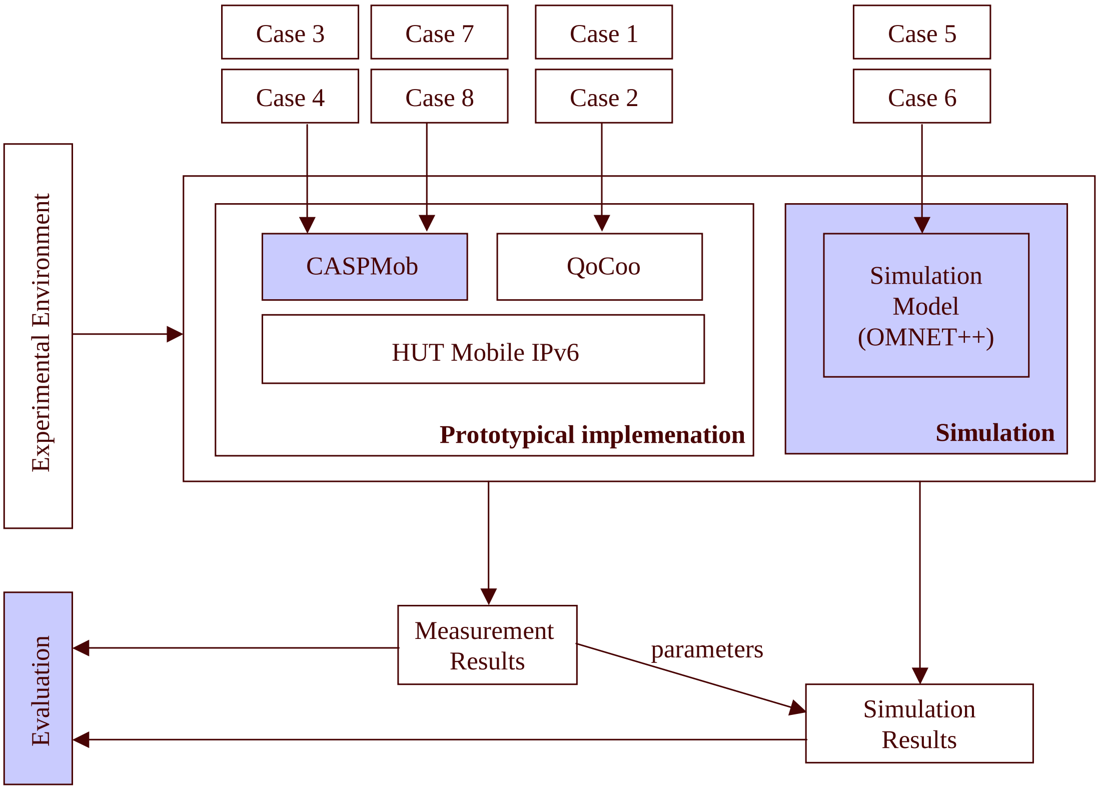


Figure 9.1: An Overview of the Methodology for Experimental Evaluation

The measurement results and simulation results are the inputs for evaluation. For the measurement evaluation, mean response time (i.e. handover latency) used as the metric; for the simulation evaluation, in addition to mean response time, two more metrics are used - number of tasks in AAAL per second and queue length at AAAL. Note that the measurement results provide necessary parameters for the simulation.

In the following sections, the shaded parts in Figure 9.1 are discussed. The validation of the CASP Mobility Client Protocol in an implementation is described in Section 9.2; the simulation environment is presented in Section 9.3; the results and discussions of the evaluation are discussed in Section 9.4.

9.2 Implementation of CASP Mobility Client Protocol

This section will first introduce the hardware and software environments of the implementation. The hardware environment has been modified and improved to the *QoCoo* testbed; the software environment includes *Linux kernel modules*, *prof-file-system* and *Netfilter system*.

Then the validation of the protocol will be described, including enhanced advertisement, QoS information based handover criteria, a modular structure of QoS, mobility and security information, the crypto aspects (e.g. cookie generation and verification, signature generation and verification, and encryption and decryption), and DoS attacking behavior.

Since total response time (i.e. handover latency) is the only metric in the experimental evaluation, the security process (re-authorization and re-authentication) has no contribution to the metric according to the definition in Section 8.1.7 despite of the speed of the re-authorization. Therefore, only the handover of scenario shown in Figure 5.20 was implemented.

9.2.1 Hardware Environment - Testbed Description

A prototypical implementation has been developed based on a MIPv6 prototypical implementation [48]. Some modifications and improvements have been done to the hardware environment of the *QoCoo* testbed[67]. Since Neumann's testbed did not consider enhanced advertisements and attacking behaviors, the main modifications to the testbed include separating the functionalities of AR1 and AR2 into two individual machines and adding a machine as an attacker.

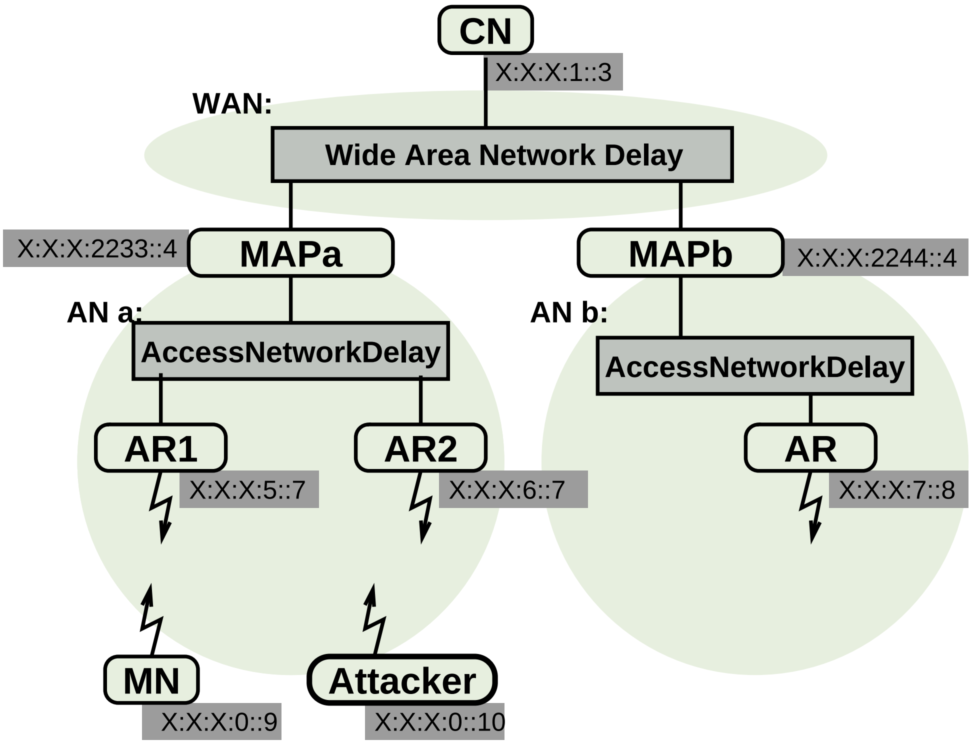


Figure 9.2: A Simplified Illustration Of The Testbed Setup

Figure 9.2 shows a simplified testbed setup from a IPv6 point of view. The IPv6 address of each entity is given in the dark grey boxes. For HA, MAP, ARa1, ARa2, and ARb, the address is advertised by their periodical advertisements. For MN and CN, the displayed addresses are the HoA. A WAN-delay simulator is configured to add the transmission delay to all IPv6 packets traveling between a MAP and the HA or CN. Similarly, a An-delay simulator is configured to introduce the transmission delay in the two access networks.

To accomplish the delay of IPv6 packets, they are routed through an IPv4 tunnel via an additional PC serving as IPv4- WAN- and AN-emulator.

In the access network of MAPa, we have two ARs connected with the MAP via the AN-delay simulator in order to emulate the enhanced advertisement concept; two mobile nodes (i.e. one normal MN and an attacker) are connected to the access network via a manageable HUB. The attacker is deployed to simulate the DoS attacks by sending bogus requests.

A detailed description of the testbed setup is shown in Figure 9.3. The Linux PCs are named as HA, MAP, CN/WAN, ARa1, ARa2, ARb, MN and Attacker.

The smaller, black-bordered boxes represent network interfaces, where those attached from the outside represent real physical interfaces and those in the inside virtual, tunnel or dummy interfaces. All interfaces are illustrated with an abbreviation for its type. If the interface is IPv6-enabled, its most relevant IPv6 address is depicted in the dark grey box. If the interface is IPv4-enabled, its IPv4 address is given in the light grey box.

All computers are Intel Pentium II machines with a CPU speed between 300 and 500 MHz. All Network Interface Card (NIC)s are 100BaseT 3Com cards. Because link b and f employ only a 10Mbps hub, the bandwidth of these links is reduced to 10 Mbps.

IPv6-in-IPv4 Tunnel

Totally, three tunnels have been installed between the MAP and the access routers. They are forwarded through the WAN emulator, which introduces WAN characteristic data traffic behavior to the messages. In the current setup the messages are forwarded through interface eth1 of the WAM emulator. The interface eth0 is used for the traffic between the MAP and the Home Agent.

The routes from the MAP to the access routers are configured manually. They have been added to the static routes file on the affected machines. Two entries have been added for the ARa2 in the MAP, one for the tunnel interface (3ffe:b80:44c:2211::8/128) and the second for prefix of the network connected to eth1 (3ffe:b80:44c:6::/64). They were included in the file */etc/sysconfig/static-routes-ipv6* in the MAP. In the other direction, the entries for the MAP have been appended in the ARa2.

The names of the computers were assigned and registered in the */etc/hosts* file in any machine, both for the IPv4 and the IPv6 address.

All the interfaces which are connected to the physical link f need an appropriate IPv4 address because of the WAN emulator. All addresses have been chosen from the X.X.2.0/24 subnet. The IPv4 addresses are part of the configuration of the tunnel inter-

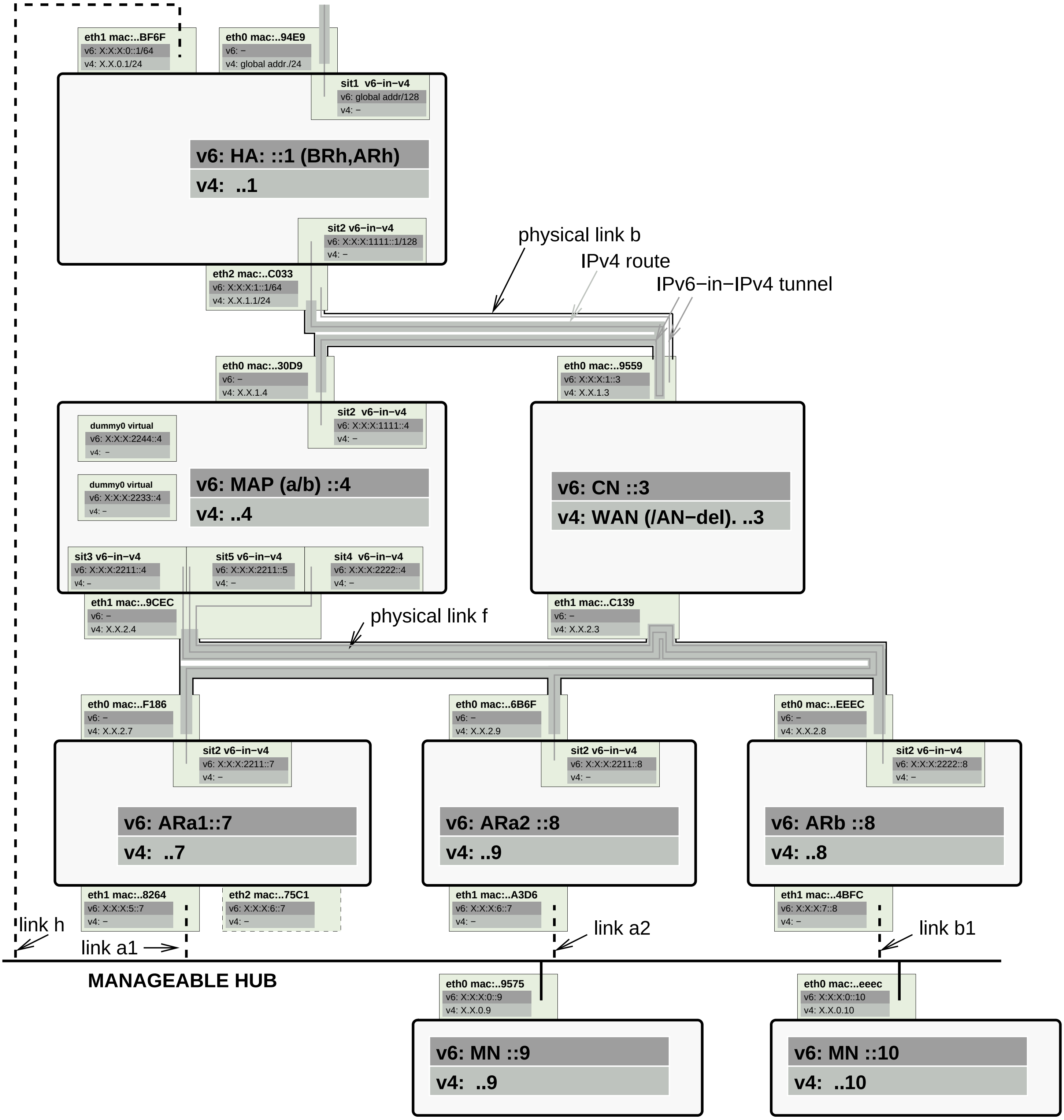
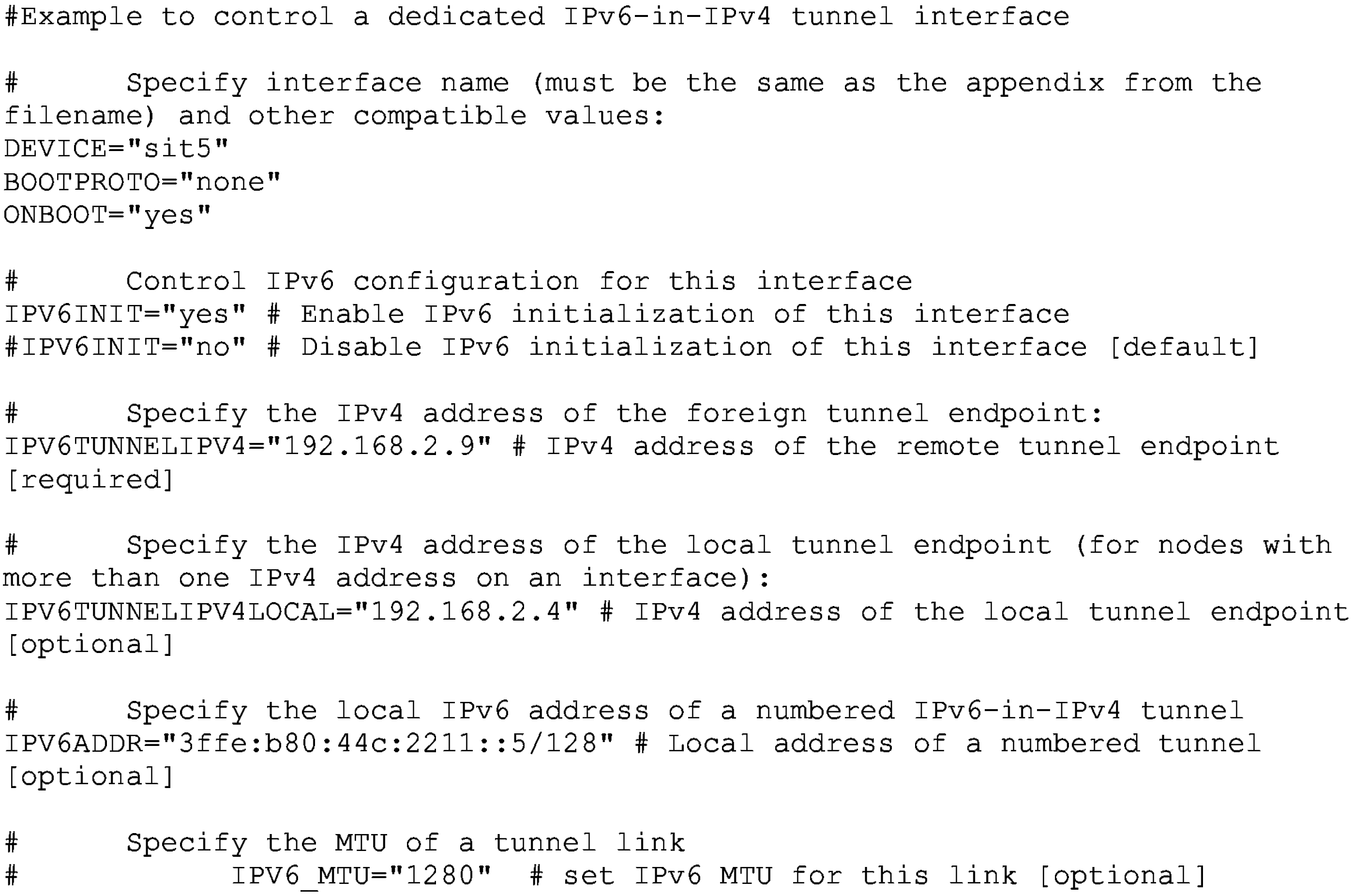


Figure 9.3: Testbed Setup

faces and directly included in the configuration files. Figure 9.4 shows the ifcfg-sit5 file from the MAP as an example.

The interfaces from the access routers, which are connected to the manageable hub



```
#Example to control a dedicated IPv6-in-IPv4 tunnel interface

#      Specify interface name (must be the same as the appendix from the
filename) and other compatible values:
DEVICE="sit5"
BOOTPROTO="none"
ONBOOT="yes"

#      Control IPv6 configuration for this interface
IPV6INIT="yes" # Enable IPv6 initialization of this interface
#IPV6INIT="no" # Disable IPv6 initialization of this interface [default]

#      Specify the IPv4 address of the foreign tunnel endpoint:
IPV6TUNNELIPV4="192.168.2.9" # IPv4 address of the remote tunnel endpoint
[required]

#      Specify the IPv4 address of the local tunnel endpoint (for nodes with
more than one IPv4 address on an interface):
IPV6TUNNELIPV4LOCAL="192.168.2.4" # IPv4 address of the local tunnel endpoint
[optional]

#      Specify the local IPv6 address of a numbered IPv6-in-IPv4 tunnel
IPV6ADDR="3ffe:b80:44c:2211::5/128" # Local address of a numbered tunnel
[optional]

#      Specify the MTU of a tunnel link
#            IPV6_MTU="1280"  # set IPv6 MTU for this link [optional]
```

Figure 9.4: An Example to Control a Dedicated IPv6-in-IPv4 Tunnel Interface

do not need a IPv4 address for communication. Every data transfer uses version 6 of the Internet protocol. The link a2 to the hub was formerly connected to ARa1 and is now attached to the new access router.

At the moment no direct connection between the access routers is required and therefore no tunnel interfaces have been installed between them.

The WAN-delay is enforced to all IPv4 packets, which are sent or received from interface eth2 of the HA or eth0 of the MAP and which are routed via interface eth0 of the machine WAN. From IPv6 point of view, the tunnel route provided by the sit2 interfaces on the HA and MAP is the only possible packet exchange route between HA and CN on one side and the MAP on the other side of the modeled wide area network. In this setup the HA also serves as ER for the CN which is physically located in the same machine as WAN and AN emulator. However, since only the CN’s interface eth0 is IPv6 enabled and the HA is the its only known IPv6 default router, all IPv6 packets leaving the CN are routed via the HA. The IPv6 layer of the CN does not know about the IPv6-in-IPv4 tunneled packets.

The machine MAP serves as MAP with two different address spaces and tunnel interfaces virtually like two MAPs for AN a and b. Therefore, in addition to the available aggregate bandwidth value and the address of the CASP node, the enhanced advertisements is configured to advertise the availability of MAP (1) with the RCoA-address space X:X:X:2233::/64 via interface sit3 and the availability of MAP (2) with the RCoA address space X:X:X:2244::/64 for AN b via interface sit4.

The AN-delay between the MAP and the ARa1, ARa2 and ARb is achieved in the same way as the WAN-delay described above. None of the physical interfaces connected to link f is enabled for IPv6. For AN a, all packets traveling between the ARa and the MAP are IPv6-in-IPv4 tunneled and routed via the interface eth1 of machine WAN. The machine WAN then delays the packets for a certain amount of time (AR-delay). Finally the ARa serves as AR for its two interfaces eth1 and eth2.

Emulation of MN's Handover

The connection of the MN to the access links of HA, ARa1, ARa2, or ARb is controlled via an SNMP manageable hub. All ports of the hub can be disabled and enabled through certain SNMP commands. When, for example, an overlapping movement from AR a eth1 to AR a eth2 is desired the corresponding ports are turned on and off respectively. In the following the MN's link-connection state will be given as a vector where the terms h, a1, a2, and b1 stand for the home, ARa1, ARa2 and ARb1 link. These terms are followed by a ":0/1" indicating whether the connection to this link is disabled or enabled by the controlling hub. One disadvantage of the manageable hub used in this testbed is the indeterministic duration (between 2 and 6 seconds) it takes from sending the SNMP command until the desired action is finally executed by the hub.

An alternative approach to emulate movements is provided by the Netfilter module. The basic idea is to let the MN physically always be connected to all potential AR interfaces, but drop packets from certain interfaces based on their MAC address. If the packets are dropped before they are further processed by the IPv6 layer, the MN has no means to deduce the connection to this particular link of the AR. By way of this, tools utilized for performance evaluation can immediately trigger a handover and have the possibility to consider for example the time when a handover was executed with a much better accuracy. Also it becomes possible to increase the maximal handover rate. This concept of a "virtual manageable hub" is employed as a software module in the MN.

9.2.2 Software Environment

The main concepts used as the software environment in the implementation include:

- **Kernel modules:** The Linux kernel enables some functionalities in modules. Thus the operating system can provide e.g. mobility support by loading a mobility module. Therefore, the functionalities of security and QoS can be enabled in modules which can be loaded and unloaded easily, without having to recompile the complete kernel during a debugging phase.

- **Proc-file-system:** The proc-file-system exists in the memory of the Linux system. It can be accessed by both the kernel modules and the use space program. Thus the proc-file-system can be used as an interface between the kernel and the use space.

- **Netfilter-system:** Netfilter [82] is used to enable the functionality of the CASP operations. It is a module for packet mangling, outside the normal Berkeley socket interface. It introduces a series of hooks at five well-defined points while a packet traverses across the IPv4 and/or IPv6 protocols stack of an IP node. Whenever a packet being processed is passed along any of these hooks, it is handled to the Netfilter module.

IPv6 and Mobile IP for Linux (MIPL) Environment

The basic IPv6 kernel provides all the functionalities necessary for processing an IPv6 packets between a network device and applications in user space, such as basic routing, Neighbor Discovery (ND) and extension header processing [67]. However, adding a UDP payload in a kernel model is not available.

In the kernel space, in order to send IPv6 packets including IPv6 header and extension headers, an IPv6 RAW socket can be allocated and and passed to the *ipv6_build_xmit()* function which assembles the packets and processes them further.

The interaction between the “Mobile IP for Linux (MIPL) Environment” and the basic IPv6 implementation is shown in Figure 9.5 [67].

All the related MIPv6 functionalities were implemented in kernel space [48] in order to reduce overhead and other complexities when interacting with IPv6 stack. The functionalities for HA, CN and MN are provided via modules which can be loaded and unloaded dynamically in the kernel.

IPv6 and Netfilter System

The position of the IPv6 related hooks and their names are shown in Figure 9.6.

Every incoming IPv6 packet which is delivered from the link-layer to the IPv6 layer first encounters the NF_IP6_PRE_ROUTING hook. Then it enters the first routing part of the IPv6 stack, which decides whether the packet is destined for another node, or a local process. If it is destined for the node itself, it is hooked by NF_IP6_LOCAL_IN before being passed to the next higher layer processing. Otherwise, if it is destined to another node instead, the Netfilter module is called again via the NF_IP6_FORWARD

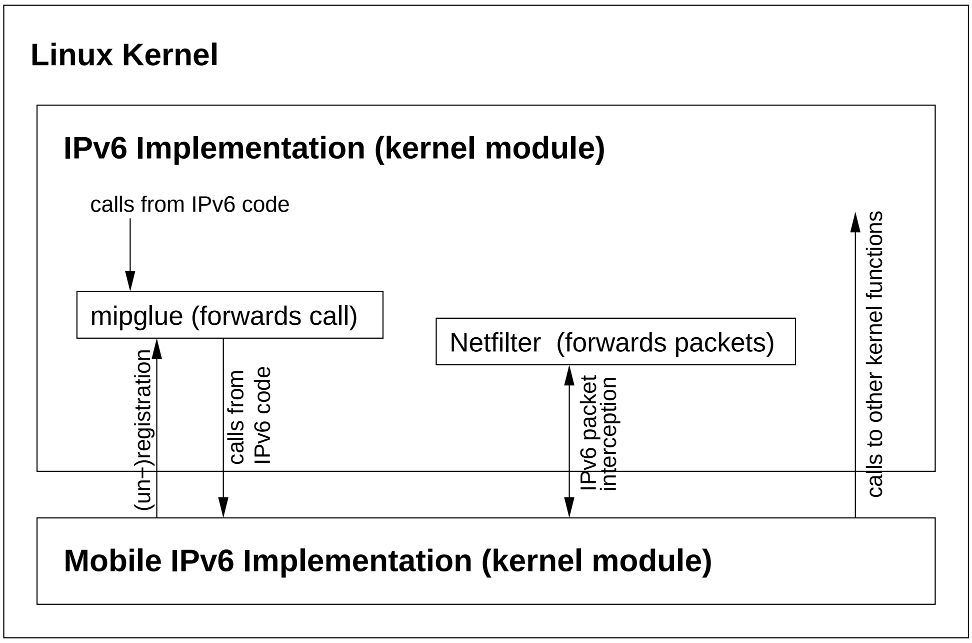

Linux Kernel
IPv6 Implementation (kernel module)
calls from IPv6 code
mipglue (forwards call)
Netfilter (forwards packets)
calls to other kernel functions
(un−)registration
calls from IPv6 code
IPv6 packet interception
Mobile IPv6 Implementation (kernel module)

Figure 9.5: MIPL General Architecture

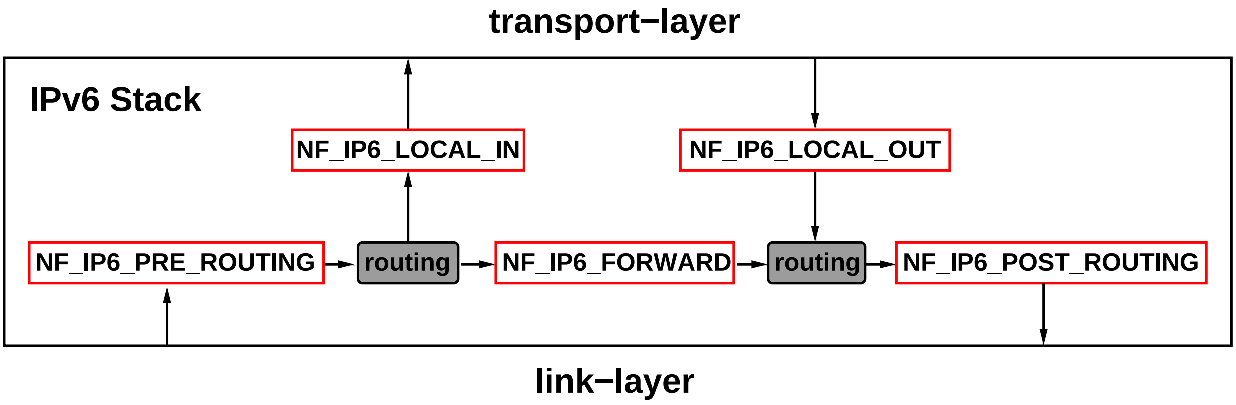


Figure 9.6: Position of Netfilter Hooks in IPv6 Stack

hook. The packet then passes a final hook, the NF_IP6_POST_ROUTING hook, before being passed to the link layer again. The NF_IP6_LOCAL_OUT hook is called for packets that are created locally before being processed by the routing part of the IPv6 stack. Other kernel modules can register to access to any of the defined hooks and are thereby provided with the possibility to modify or even drop such intercepted packets.

9.2.3 Implementation of the Enhanced Advertisement Mechanism

In Linux operating system, broadcasting advertisement is achieved by the router advertisement daemon (radvd) [81]. It is a program running in user space at a Linux IPv6 router. The enhanced advertisement daemon which is based on it is also implemented in the user space. Whereas, the other developed concepts of the proposal are implemented in the kernel space in order to interact well with the existing MIPv6 models which are running in kernel. Therefore, sockets should be well defined for the communication between the enhanced advertisement daemon and the kernel models.

The advertisement daemon from [81] has been modified to "MAP Discovery" in a HMIPv6 environment [68]. The MAP discovery service includes advertising of a MAP's network information. However, no bandwidth information is available in the advertisements. To enable the enhanced advertisements, the available bandwidth at a node should be modified dynamically by the kernel mobility and QoS models; the aggregate bandwidth information should be broadcasted by ARs.

In the testbed, only one-layer MAP has been implemented. The advertisement daemons for sending advertisements at MAP and AR run in the user space.

MAP obtains its own bandwidth value from a proc file - $mipv6\_ad\_bw$. It adds the value in the "aggregate available bandwidth" field and its IPv6 address in the "next CASP node" field. Then it broadcasts the advertisement via all of its interfaces to the ARs. The interval of the advertisements can be controlled via the file "/etc/radvd.conf" shown as Table C.7.

ARs cache the received "the Enhancement Option" in its $other\_map\_list$ structure. When it is going to send out its own advertisement, it extracts related information (including MAP's IPv6 address and bandwidth) from the MAP option in the $other\_map\_list$, and puts them into its own advertisement. Then it obtains its available bandwidth value from its proc file (i.e. $mipv6\_ad\_bw$) and compares it with the bandwidth value from the MAP option. If its own value is greater, it makes no modification to bandwidth value; otherwise, it replaces the bandwidth value with its own available bandwidth value. Since all the ARs are assumed to be CASP node, AR fills the "next CASP node" field with its own IPv6 address. The interval of the advertisement from AR can also be controlled via the "/etc/radvd.conf". The sending rates of advertisements from MAP and AR are independent. Normally the sending rate of advertisements from AR is configured to be faster since MAP is assumed to have larger bandwidth capacity than ARs.

Once a reservation is made, MAP and AR need to update the available bandwidth information in the $mipv6\_av\_bw$ proc file accordingly. The dynamic change of the bandwidth value in the proc file is done by the mobility module in the kernel.

9.2.4 Implementation of Handover Criteria

The MN processes the received advertisement in kernel. Recall from the specification chapter, Receipt of an advertisement is one of the triggering events to invoke MN's operations. Generally MN extracts the aggregate bandwidth information and other related information and makes a handover decision based on a handover criteria as shown in

Figures 5.7 and 5.8.

As discussed in Section 5.2.1, a set of handover state of handover (i.e. "lazy" or "eager") has been implemented via the proc-files mn_ho_state.

9.2.5 Modular Structure

Since the modules of the CASP Mobility protocol are carried as UDP payload in a packet, the existing UDP related functions which were written in c for user space calls should be modified accordingly for the calls from the kernel space.

In the kernel space, A new Netfilter module namely qos-monitor is created to intercept the traversing messaging layer and client layer information. Figure 9.7 shows the responsibilities of CASP related processing at different entities.

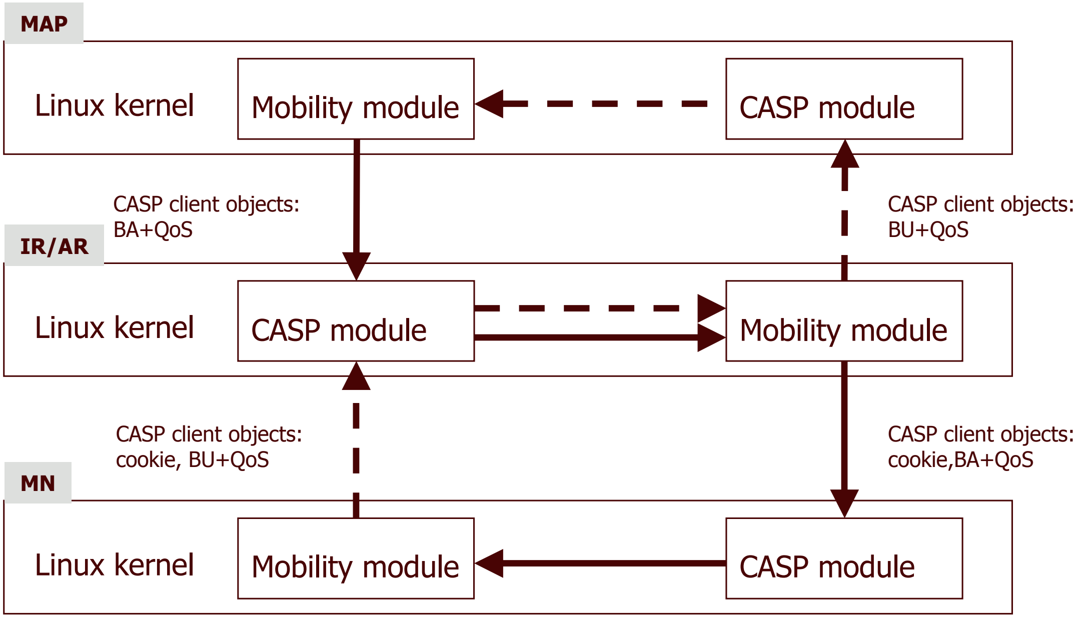


Figure 9.7: Responsibilities of CASP Related Processing

Since the existing mobility functions were implemented in the Linux kernel, the CASP functionalities are also implemented in Linux kernel. Because of the end-to-end communication between every two nodes, mobility mode at each node is responsible for sending a packet while the CASP module is responsible for receiving a packet. The dotted line represents the CASP QUERY message flow whereas the solid line is the CASP RESERVE message. On the testbed, only one layer of intermediate router (i.e. access router) is implemented between MN and MAP. Therefore, MAP is also the crossover router in any mobility cases.

9.2.6 Crypto Aspects

The crypto aspects include cookie generation and verification, signature generation and verification and encryption and decryption with $TmpKeyDistKey_{AN}$ as shown in Figures 5.19 and 5.20.

Figure 9.8 presents an example result of cookie generation. The example shows that a typical duration to generate a cookie is 37 μs. The hash code is HMAC-SHA1.

```
mn_id=1234 generator_id=abcd
creation_time: 3d8ae457
random number: 2c1e4ccc
cookieKey: 54 5 31 31 c3 c6 94 66 1b 4f 9f 18 8a 97 92 98 33 3b ca b
hash code: 27 b0 a 75 56 e8 1b ad b6 32 90 97 7a 73 8e 86 a5 46 ad 7c
Duration to generate a cookie: 37
```

Figure 9.8: An Example Result of Cookie Generation

The pseudo-code of cookie verification is shown in Figure 9.9.

The parameters for a temporary IPSec SA in the direction from MN to AR are carried in the CASP Client objects as UDP payload. The operations of signature generation and verification use HMAC-SHA1 and the operations of encryption and decryption with $TmpDistKey_{AN}$ use triple DES [84]. The source code of the triple DES was obtained from [1].

9.2.7 Emulation of DoS Attacks

The flooding behavior was implemented justify the robustness of the schemes shown in Table 9.1 against Denial of Service (DoS).

In case that DoS attacks exist, an MN computer emulates an attack by generating and sending "wrong cookies" [1] at various rates.

9.3 Simulation Environment

Figure 9.10 shows a topology overview in the simulation.

The simulation model consists of one AAAL, one MAP, eight ARs eight mobile environments (MOBENs). The AAAL connects the MAP with a 100 Mbps Ethernet link; the MAP connects every ARs with 100 Mbps Ethernet links; Each AR is responsible for one MOBEN, which represents 4000 MNs and one attacker. Each MN or the attacker

[1] wrong cookies are deemed to cause cookie verification failed.

```
loop{wait for packet received from link layer}
    Parse cookie object from hop-by-hop header
       Get information: mn_id, cookie generator_id,
           creation_time, random_number, hash_code
       Check creation_time
         if expired then
            Drop packet
            Break
       Check cookie generator_id
         if not on the trusted list
            Drop packet
            Break
       Compute a hash code (HMAC_SHA1)
       Compare the result to hash_code
         if not identical
            Drop packet
            Break
       Start QoS+BU and security processes
       Notify the ARs on the trusting list about the cookie
   end if
   else if Packet indicates inter-domain handover then
       Perform inter-domain handover process
   end if
 end loop
```

Figure 9.9: The Pseudo-code of Cookie Verification

sends requests to the corresponding AR as re-registration requests via a wireless link of 11 Mbps data rate.

At each AR, the inter-arrival traffic from legitimate MNs is modeled with a Poisson process with the mean inter-arrival time of 0.5s [36]; Each of the 4000 MNs in one MOBEN performs 1.8 intra-domain handovers per hour.

The traffic of an attacker is modeled with deterministic process. That means an attacker always sends bogus requests with a constant rate.

9.4 Evaluation Results and Discussion

The evaluation results and discussions of measurement and simulation are given in Section 9.4.1 and Section 9.4.2 respectively.

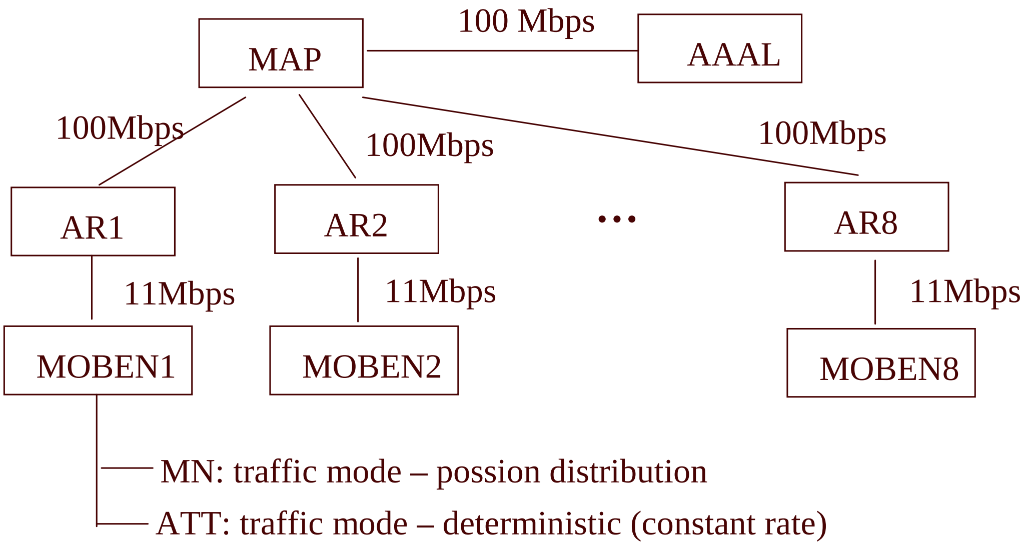

MAP
100 Mbps
AAAL
100Mbps
100Mbps
100Mbps
AR1
AR2
...
AR8
11Mbps
11Mbps
11Mbps
MOBEN1
MOBEN2
MOBEN8
MN: traffic mode – possion distribution
ATT: traffic mode – deterministic (constant rate)

Figure 9.10: Topology Overview In the Simulation

9.4.1 Measurement Results and Discussion

Parameters of processing times is given first. Then the measurement results of total response time are presented.

Parameters of Processing Times

To provide the parameters of processing times to the mathematical analysis and simulation, measurements are taken using *tcpdump* [2], monitoring and recording the times when an operation starts and ends. Table 9.2.

Operation	min	max	average	s^2
Generating an AA request	150.650	155.006	152.457	0.00340
Forwarding an packet	15.998	24.137	20.189	0.00669
Authorizing a request	148.331	153.413	151.814	0.00294
Reserving resources	201.365	245.424	219.608	0.01391
Generating/verifying a cookie	36.956	39.261	37.910	0.00226
Encryption/decryption	39.781	41.359	40.618	0.00330
Session key lookup	12.497	22.087	18.139	0.01462

Table 9.2: Processing Times Obtained From Measurements (μs)

[2] *tcpdump* is a tool to capture and dump network packets in order to make statistical analysis. It makes use of the packet capture library *libpcap* [94].

Total Response Time

The total response time for a re-registration request of a MN is the difference in times between departure instance of a registration request message (i.e. CASP QUERY message in the CASP cases) and the arrival instance of a registration reply message (i.e. CASP RESERVE message in the CASP cases). The latency caused by ICMPv6 neighbor discovery for the next hop address is included in the response time.

Six cases are studied in the measurement evaluation:

- **Case 1**: AR first verifies a cookie when receiving a re-registration request. If the check passes, it starts the QoS-conditionalized BU process. The re-registration request packet is destined to MAP. The cookie is included in the Hop-by-Hop option in the IPv6 header of the packet. The AR notifies the use of the cookie to the ARs within the cookie's "Area of Validity" while starting the QoS+BU process. When MAP receives the QoS+BU message, it communicates with AAAL for security checks. When it gets a positive answer from AAAL, it continues the QoS+BU process, acknowledging the operations of resource reservation and binding update.

- **Case 2**: Without an security check, AR starts immediately a QoS-conditionalized BU process, reserving resources upon the request. When MAP receives the QoS+BU message, it communicates with AAAL for security checks. If the re-AA fails, AR drops the packets silently without notifying MN. When MAP gets a positive answer from AAAL, it continues the QoS+BU process, acknowledging the operations of resource reservation and binding update.

- **Case 3**: AR first verifies a cookie when receiving a re-registration request. If the check passes, it starts two processes (i.e. QoS+BU and re-authorization processes) in parallel. The QoS+BU process is performed using CASP Mobility Client protocol. The re-registration request packet is destined to AR. After the cookie verification, AR generates a new packet and send it to MAP. The cookie is included in a CASP client object in the UDP payload. The AR notifies the use of the cookie to the ARs within the cookie's "Area of Validity" while starting the QoS+BU and re-authorization processes. It is assumed that the result of the re-authorization process arrives earlier than that of the QoS-conditionalized BU process so that the time spent on the former process has no contribution to the contribution to the re-registration response time.

- **Case 4**: AR generates an AA request message and sends it to AAAL for an authentication and authorization check when receiving a re-registration request. When it receives a positive re-AA ACK message for the request from AAAL, the

AR starts a QoS+BU process for the request, reserving resources upon the request and performing an binding update operation at MAP. The QoS+BU process is performed using CASP Mobility Client protocol. If the re-AA fails, AR drops the packets silently without notifying MN.

- **Case 7**: Case 7 is Case 3 in the presence of DoS attacks.
- **Case 8**: Case 8 is Case 4 in the presence of DoS attacks.

Table 9.3 lists the average, minimum and maximum registration delays in the six cases (i.e. Cases 1, 2, 3, 4, 7, 8. Cases 5 and 6 are discussed in Section 9.4.2.) after a total of 100 different reading were taken in each case. The total controlled transmission delay applied to the registration messages by the AN simulator is given in Column *delay*. Since the compared cases are intra-domain handover cases, only the AN delay comes into play. The variance is given in the column s^2.

Case No.	delay	min	max	average	s^2
1. QoSCondiBU + cookie	20	21.542	23.780	22.400	0.00459
2. QoSCondiBU + AAA	20	21.193	24.327	22.223	0.00386
3. CASPMob + cookie	20	22.317	24.413	23.345	0.00416
4. CASPMob + AAA	40(120)	68.933	366.745	253.585	0.011751
7. CASPMob + cookie + DoS	20	22.277	27.025	23.908	0.03484
8. CASPMob + AAA + DoS	40(120)	213.099	993.397	528.778	0.0391

Table 9.3: Experimental Results for Total Response Time in Different Handover Scenarios (ms)(WAN delay: 40ms, AN delay: 10ms).

Since there is only one round-trip signaling between MN and MAP in Cases 1, 2, 3 and 7, 20 ms is added in the registration delay as the transmission delay; Cases 4 and 8 have two round trips (i.e. first one is the AAA process and the second one is the QoS and BU joint process), 40 ms is introduced. When HA needs to be involved for an AAA check, additional 80 ms is taken into account since WAN simulator is set to 40 ms.

The experimental results show that the measured signaling latencies mostly depend on the delays enforced by the WAN and AN simulator. Delay caused packet processing in the involved nodes is only of minor importance.

By comparing Cases 1 and 2 in which the QoS-conditionalized BU approach is used to establish the new QoS path and perform the mobility management, we conclude that the cookie mechanism increases the registration latency by 0.80 percent due to the corresponding operations. The increased 0.177 ms could be negligible to any applications.

The comparison of Cases 1 and 3 shows that the CASP Mobility Client Protocol increases the registration latency by 4.2 percent. This is due to the fact that end-to-end communication is enabled between the adjacent two hops whereas the access router is the intermediate router in the end-to-end communication between the MN and the MAP. The feature of more scalability of the CASP Mobility Client Protocol has the cost in the respect of registration latency.

Since in Case 3 the QoS and BU joint process does not wait for the result from the AAA process as in Case 4, the optimization of the cookie mechanism reaches 986.2 percent. If the home domain is involved for an AAA check as in Case 4, the MN can continue its session on the new QoS path in Case 3 without waiting for the AAA result and the user data can be protected by the temporary SA. Therefore, we conclude that the CASP Mobility Client Protocol is more efficient in QoS-aware micro-mobility handover cases.

In the presence of DoS attacks shown in Cases 7 and 8, the attacking rate is 10 times as much as the arrival rate of normal MNs. When the cookie mechanism is deployed (i.e. Cases 3 and 7), the registration latency increases 2.4 percent whereas in the no-cookie Cases (i.e. Cases 4 and 8), the registration latency increases 108.5 percent. Therefore, we conclude that the cookie mechanism is a secure and efficient scheme in preventing DoS attacks.

9.4.2 Simulation Results and Discussion

To evaluate how efficient the cookie mechanism works in a re-registration process, two more metrics are used in the evaluation in addition to the mean response time - number of tasks in AAAL per second and queue length at AAAL:

- **Mean response time**: the duration between the transmission of the first bit of a re-registration request of a legitimate MN and the arrival of the last bit of the corresponding re-registration response. This parameter can reflect the signaling capacity of a path the efficiency of an intra-domain handover.

- **Number of tasks in AAAL per second**: how many re-AA tasks in Case 0 and how many re-authorization tasks in Case 1. When AAAL has too high throughput of re-AA, the computing capacity is threaten by DoS attacks.

- **Queue length at AAAL**: how many messages waiting in the incoming queue for being processed. When AAAL has too many messages waiting in its queue, its storage capacity is being exhausted.

Cases 5 and 6 are studied in the simulation evaluation. They are equal to Cases 1 and 2 in the presence of DoS respectively.

Figure 9.11 is given for illustrative purposes and shows how the attacking rates increases over time during the simulation. The starting point of the attacking rate is set to

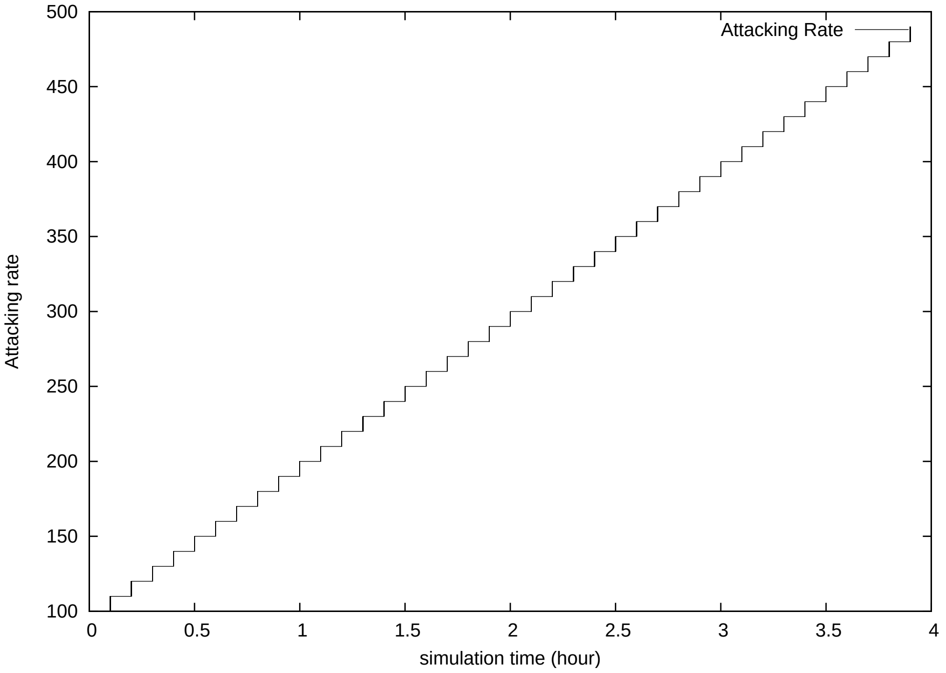


Figure 9.11: Increase of Attacking Rate Over Time

100, and the attacking rate increases by 10 every six minutes. During each six-minute period, the attacking rate is constant.

Figure 9.12 shows how the increasing attacking rate influences the average re-registration delay. Before the attacking rate reaches 200 per second, the average re-registration delay stays relatively constant. That means attackers does not give the access network any trouble with the attacking rate less than 200 per second.

Afterwards, with the increasing of the attacking rate, the average re-registration delay in no-cookie case (Case 0) goes up gradually while the parameter in cookie cases (Case 1 and Case 2) still keeps stable. This proves that the cookie mechanism is efficient in preventing depletion of signaling capacity and beneficial of optimized intra-domain handovers.

Figure 9.13 shows that the cookie verification at ARs can prevent AAAL from heavily loaded by re-authentication and re-authorization works, and prevent AAAL's queue from being filled up shown in Figure 9.14.

Therefore, this performance evaluation shows that the cookie-based mechanism is efficient to deal with DoS attacks, and therefore improves the optimized and QoS-aware handovers.

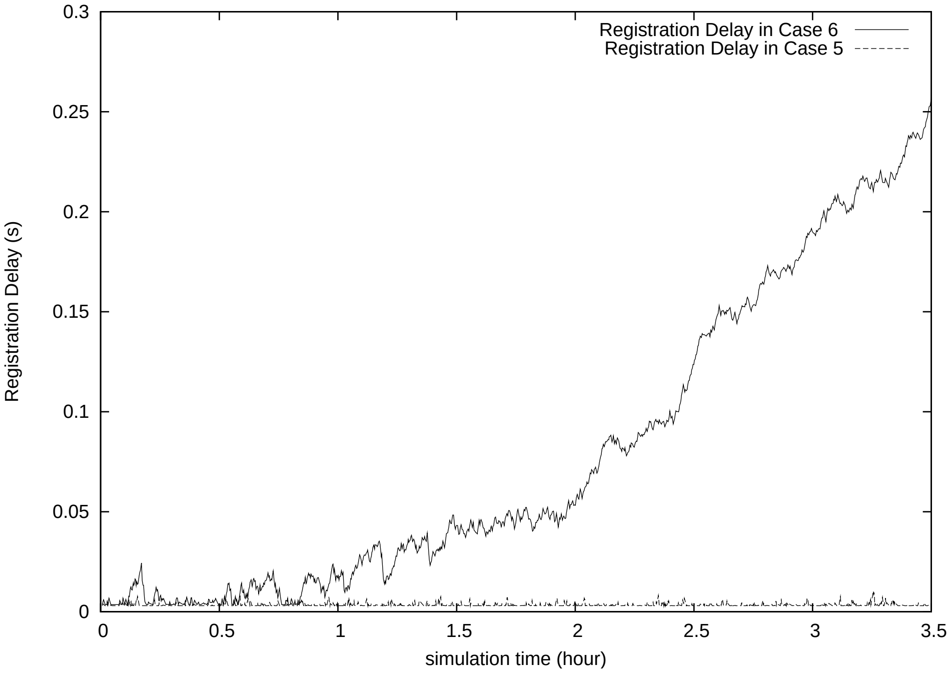


Figure 9.12: Impact of Increasing Attacking Rate on Mean Response Time

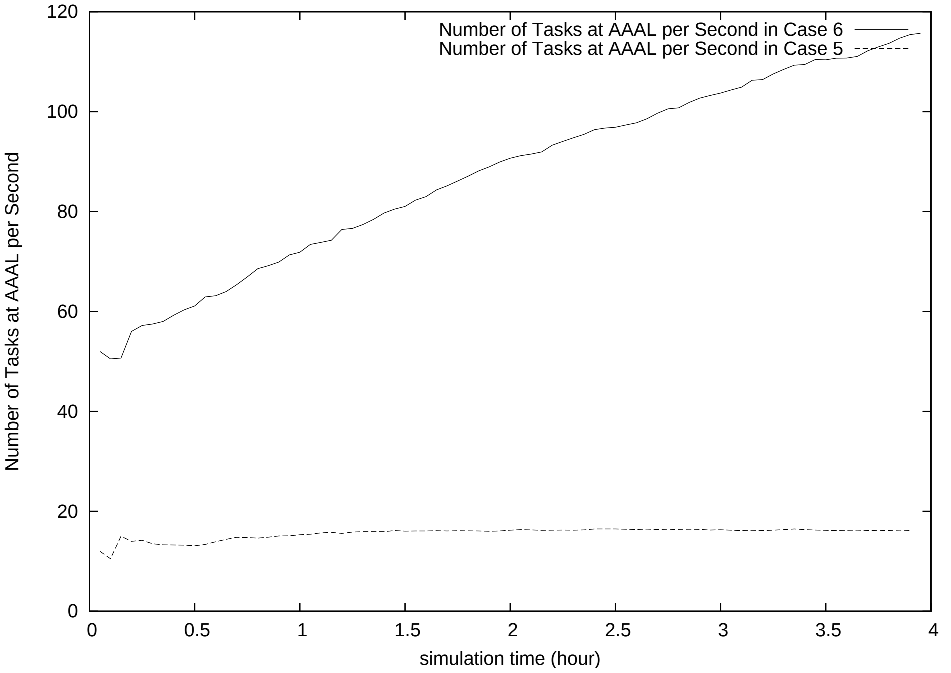


Figure 9.13: Impact of Increasing Attacking Rate on Number of Tasks in AAAL

9.5 Summary

This section described the experimental evaluation of the proposal. It aims to provide parameters in the mathematical analysis, as well as find answers for some interesting questions.

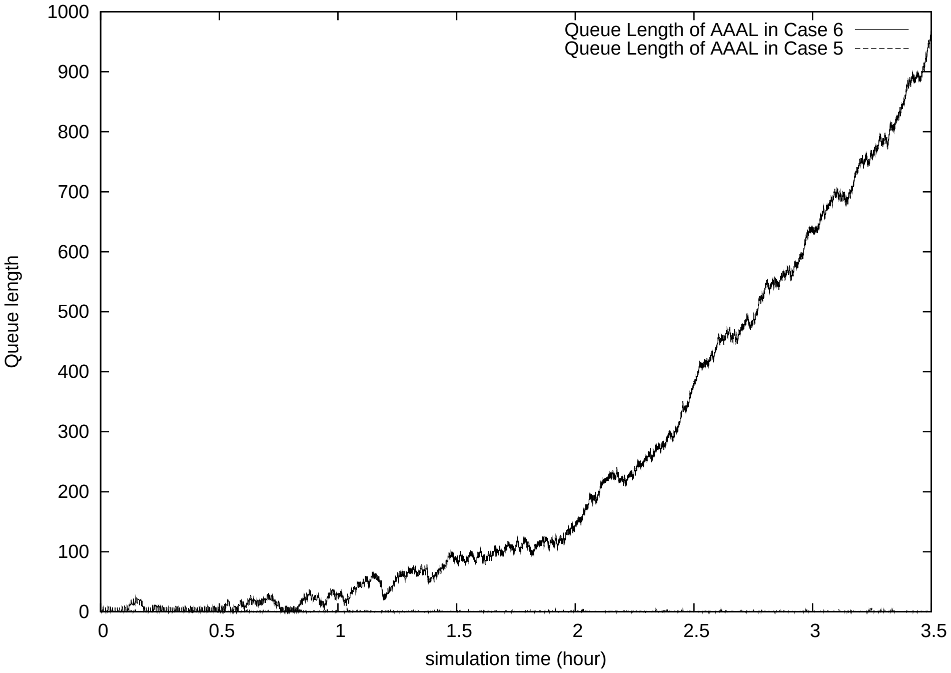


Figure 9.14: Impact of Increasing Attacking Rate on Queue Length of AAAL

To meet the goals, eight cases were selected for the performance evaluation in a measurement environment and a simulation environment. The total response time and queue length at AAAL and number of tasks at AAAL are the metrics to evaluate the formulated proposal by comparing the performance of the eight cases in terms of handover latency and robustness against DoS attacks.

The findings of the experimental evaluation are summarized as follows:

- The CASP Mobility Client protocol performs slightly worse than the QoS-conditionalized binding update approach in the specific scenario of intra-domain handover in a HMIPv6 environment.

  This happens because, in the CASP Mobility Client protocol, AR has to generate a new packet (i.e. CASP QUERY) and send it to MAP. This scheme takes longer time than the QoS-conditionalized binding update approach in which AR parses only the cookie information from the hop-by-hop option in the IPv6 extension headers.

  However, the extra time is trivial comparing with the total response time. Moreover, the characteristic of the two-layer and modular structure is regarded as more scalable in the IETF community. The validation of the CASP Mobility Client protocol will be helpful for further investigation.

  The experimental results prove that the CASP Mobility Client protocol can achieve a secure and QoS-aware micro-mobility handover efficiently.

- The experimental evaluations in both measurement and simulation show that the proposals which use the cookie mechanism outperforms obviously the other approaches in the presence of DoS attacking. It has been proved that the cookie mechanism is a method to protect against DoS attacks in QoS reservation process.
- It is observed from the measurement evaluation that in case that there is no DoS attacks in the access network, the dominating factors in a re-registration process are propagation delay and link switching time. The response time was measured on the testbed with the handover mode of "break-before-make", which means that MN performs ICMPv6 neighbor discovery for the new AR's address after it notices that it has disconnect with the its old AR. The neighbor discovery process play a pole in the total response time.

The conclusions drawn in this Chapter are in accordance with those from Chapter 8 and Chapter 7.

Chapter 10

Conclusions and Outlook

This dissertation addresses the challenge how to achieve a secure, efficient QoS-aware mobility support in IP-based networks. The dissertation first identified the importance of establishing a new QoS path for a active session in a secure and efficient way. Based on the related work discussion, we concluded that no proposal so far has addressed to achieve an optimized overall registration procedure including the mobility management, QoS path re-establishment while taking security threats into account.

The micro-mobility schemes address the mobility management optimization only; the combination of the fast handover scheme and context transfer aims to provide seamless mobility support and fast session state re-establishment, but it does not address the QoS path setup and protection against the DoS attacks.

In order to address the identified issues, a signaling protocol to achieve a secure, efficient QoS-aware mobility support in IP-based networks has been proposed in the dissertation. We include the aggregate available bandwidth value in the advertisements so as to enable MNs to select the most suited AR as the handover target. MN makes the handover decision based on both its handover mode (i.e. lazy or eager) and the network conditions (i.e. link availability and the provided bandwidth value).

In an access network with the HMIPv6 infrastructure, a cookie verification is performed at the target AR as the first-step authentication to reduce the threat of DoS attacks. The the verification passes, the AR initiates a QoS and BU joint process towards MAP. At the same time, it also initiates an authorization process towards AAAL. In case the authorization process takes longer time (e.g. due to involving AAAH for the purpose) than the other, MN is able to use the new QoS path and the user data is protected by a temporary SA. When the authorization process is complete, the AR authenticates the MN' ongoing session. If the authentication is successful, the corresponding policy can be enforced at the AR to support DiffServ; AR generates a new cookie for the MN's next intra-domain handover and starts the definitive SA to protect MN's user data.

All the operations are integrated in a re-registration procedure as shown in Chapter 5. The emphasis and the main achievement lie in the optimization of the intra-domain

HO in micro-mobility scenarios. All the measures of combining the signaling of the mobility, QoS re-establishment processes, parallelizing the QoS-aware re-authorization process with the QoS re-establishment process, performing a preliminary check with a cookie, enhancing the advertisements with QoS information and securing user and signaling data with a temporary session key aim at optimizing the intra-domain handover procedure.

Note that the cookie based mechanism and the idea of temporary SA establishment can be adapted in any forms of two-layer QoS signaling protocols. They are not limited to be applied in CASP mobility client protocol. The essence of the two concepts is minimize the latency introduced by

- deploying policies of access control, DiffServ or authorization to the new AR;
- re-authorization with AAAL, or AAAH if necessary;
- re-authentication;
- protect network resources against potential DoS attacks.

The active session is ongoing along the new QoS path while the above operations are undertaken. Therefore, we can reach an optimal tradeoff between efficient QoS-aware mobility and security achievement.

Also note that the developed concepts can be integrated with some solutions which are being developed in working groups in IETF. For example, when packet loss is essential for an application, FMIPv6 can be used with HMIPv6 to minimize the packet loss rate during the period that the new QoS path is being set up. FMIPv6 allows the packets to be transmitted from the old AR to the new AR even though these packets have no QoS guarantee. It is compatible with all developed concepts in the work. CT may also be useful to enable the communication between the old AR and the new AR for e.g. authentication and authorization state re-establishment, especially when new AR takes longer time to communicate with MAP for the purpose.

The mathematical analysis and experimental evaluations show that the proposal is successful in achieving a secure, efficient and QoS-aware mobility support in IP-based networks. The proposal is an overall solution which is deployable in a real access network.

10.1 Contributions

In summary, the major contributions of the dissertation include

- including QoS information and CASP node's address information in the advertisements;

- making handover decision based on availability of resources information in addition to the network information;
- protecting the access network against DoS attacks with a cookie-based mechanism;
- protections of signaling and data traffic in a separate manner during the phase between MN starts using the new QoS path and the registration procedure finish completely;
- including the QoS signaling and the mobility signaling in one integrated process by using a modular signaling protocol;
- optimizing re-registration procedure in local movements by parallelizing resource reservation and BU joint process and re-authorization process;
- enabling policy-based DiffServ support after a practical authorization process in the HMIPv6 architecture;
- designing possible solutions to achieve a comprehensive registration procedure including session state information re-establishment and mobility management.

10.2 Outlook

A number of interesting research topics arise from the dissertation:

- Since the wireless bandwidth is expensive, there will be irresistible pressures to improve the efficiency of the air link between the mobile node and access network [52]. Context Transfer is useful in establishing the DiffServ QoS state for the "last hop", especially when it takes relatively long time to enforce a policy at the new AR. Although the current proposal from FMIP and CT can not meet the optimization requirement in an overall registration procedure, they are useful in reducing packet loss. This feature is crucial for the applications which are sensitive to pack loss such as File Transfer. With the increase of computing power of a mobility entity, an access network will become more resistant against early saturation when applying the two schemes. Therefore, how to integrate the CT and FMIP in our proposal is one of the interesting research topic.
- It has been observed that the performance of the link layer is crucial in a handover process even though this dissertation discussed optimization of a mobility support in IP layer. An efficient interaction with link layer may have a gain in the overall performance for a secure and QoS-aware mobility support. For example, to enable the IntServ support on the wired path and DiffServ on the wireless path, a

QoS parameter must be converted to link layer operations or policy actions. The efficient conversion is preferable in realizing QoS support on a end-to-end path. This is worth further investigation.

- This dissertation addressed somehow the e-commercial issues, such as authentication, authorization and resource reservation. It will be very interesting to apply the formulated proposal in reality. Mobile users have various behaviors in an access network, which may cause some unusual situations such as authorization for a QoS request fails for a credible user due to his misuse of his credit. Furthermore, billing is also an important issue in mobile e-commerce. Investigation on this issue is one of the further steps based on the thesis.

Appendix A

Packet Formats

A.1 CASP Mobility Client PDU Structure

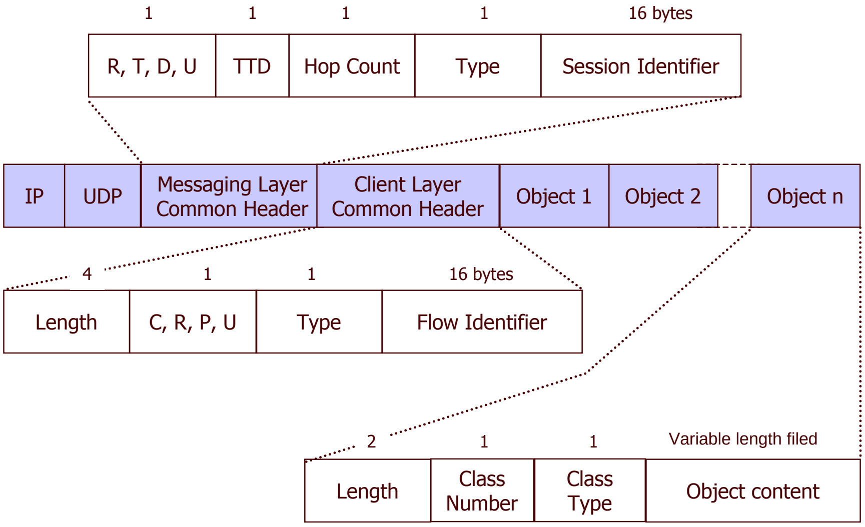


Figure A.1: CASP Mobility Client PDU Structure

The explanation of the fields in the "Messaging Layer Common Header" and the object header refers to [87]. Since UDP is used as the transport protocol, a length field in the "Messaging Layer Common Header" is necessary.

See [85] for the details of the fields of "Client Layer Common Header".

A.2 CASP Mobility QoS Objects

The field of "CASP Mobility QoS Answer Object" is identical to the QoS request object except that a "RBW" field which holds the reserved bandwidth is included rather than

0		31
Length (bytes)	Class-Num	C-Type
Lifetime		
MN Address (16 bytes)		
CN Address (16 bytes)		
MN-HoA (16 bytes)		
Desired bandwidth	Acceptable bandwidth	

Figure A.2: CASP Mobility QoS Request Object

0		31
Length (bytes)	Class-Num	C-Type
Lifetime		
MN Address (16 bytes)		
CN Address (16 bytes)		
MN-HoA (16 bytes)		
Desired bandwidth	Reserved	

Figure A.3: CASP Mobility QoS Answer Object

two bandwidth fields.

A.3 CASP Mobility BU and BA Objects

The fields of the eight bytes after the object header is identical to the definition in [43].

The field explanation of the BA object refers to that of the BU object. Exclusively, it has a “Refresh” field which means the period after which the MN needs to send a new BU request MAP. The refresh interval is smaller than the lifetime of the binding entry.

A.4 Other CASP Mobility Objects

Table A.1: Field Explanation of the CASP Mobility QoS Request Object

Field Name	Description
Lifetime	Lifetime of a QoS reseration. When the valid time of a reservation will expire soon, a MN needs to send a message to refresh the reservation.
MN Address	The upstream source of a session. This field along with the next two following fields are used by the entities along a path to define a specific mapping between MN's IPv6 packets and the session identifier for which a reservation has been made.
CN Address	The downstream source of a session.
MN HoA	An global unique identification of the MN.
DBW	Desired bandwidth. It is the upper bound of the range of a QoS request.
ABW	Acceptable bandwidth. It is the lower bound of the range of a QoS request.

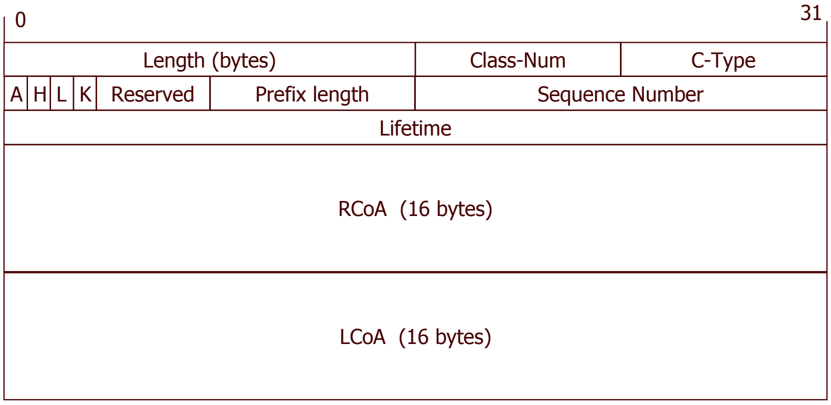


Figure A.4: CASP Mobility Binding Update Object

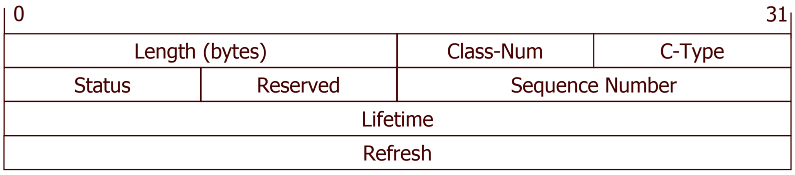


Figure A.5: CASP Mobility Binding Acknowledgement Object

Table A.2: Field Explanation of the CASP Mobility BU Object

Field Name	Description
A	If the Acknowledge (A) bit is set to “1”, a Binding Acknowledgement is required upon receipt of the Binding Update message.
H	If the Home Registration (H) bit is set to “1”, MAP should act as MN’s home agent.
L	Link-Local Address Compatibility (L) is set when the home address of MN has the same interface identifier as its link-local address.
K	If the Key Management Mobility Capability (K) bit is cleared, the protocol used for establishing the IPsec security associations between the MN and the home agent has to be rerun in case of mobility. In this thesis, this bit is also set to “1” which means the permanent trust relationship between MN and its HA.
Prefix length	the length of the prefix of a IPv6 address
Sequence Number	it is used by the MAP to sequence Binding Updates; MN use it to to match a returned Binding Acknowledgement with this Binding Update message.
Lifetime	To specify the valid peorid of the binding update
LCoA	Local link Care-of Address which is valid in the MAP domain. A Binding entry at MAP holds LCoA and RCoA.
RCoA	Regional Care-of Address which is global unique IPv6 address.

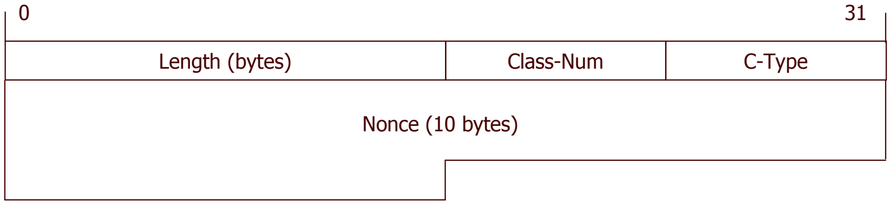


Figure A.6: CASP Mobility Nonce Object

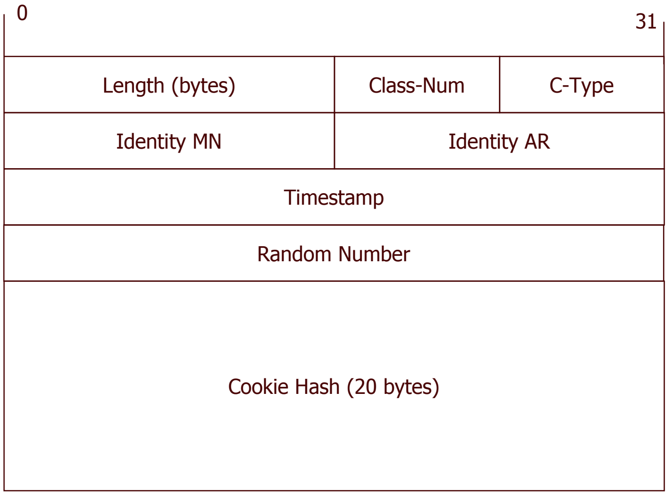


Figure A.7: CASP Mobility Cookie Object

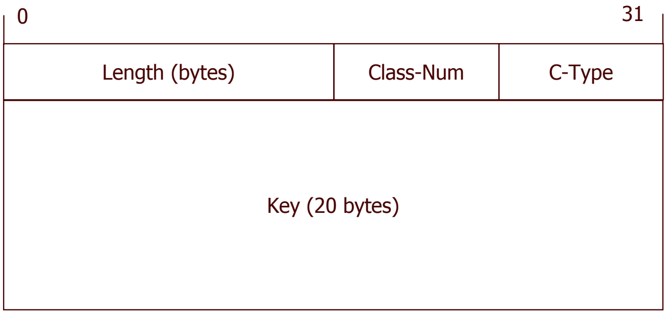


Figure A.8: CASP Mobility Key Object

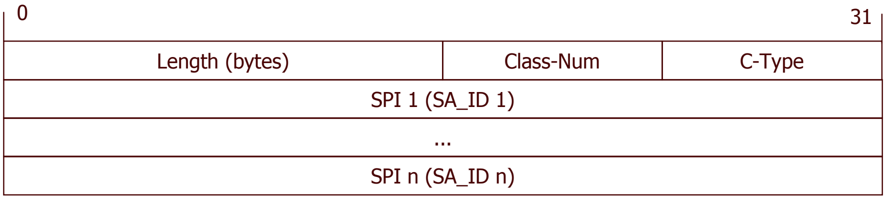


Figure A.9: CASP Mobility IPSEC SPI Object

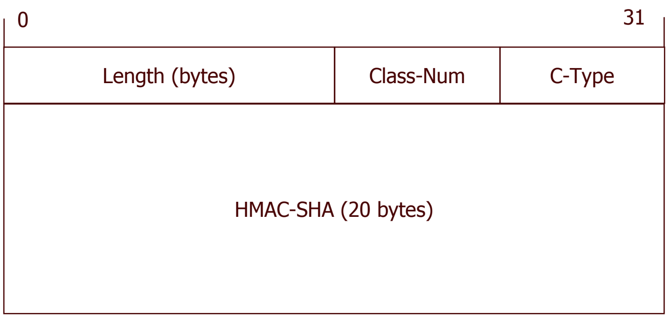


Figure A.10: CASP Mobility Sginature Object

Appendix B

Protocol Operations in case of Powerup and Inter-domain Handovers

This chapter describes the protocol operations in case of powerup and inter-domain handovers.

B.1 Protocol Operations of the Powerup Case

The power-up cases are also called “Initial Registration”, which means a mobile node visits a foreign domain and registers initially its new unicast IP address with its home domain and its corresponding node. The mobile node obtains the IP address from AR’s advertisements or by sending a solicitation request to an AR. The basic design guidelines for the protocol operations are listed as follows:

- Authentication and authorization checks should be performed before the access network takes any other actions on a MN’s request. Since the MN is “unknown” in the access network, the checks must be done with the help of e.g. its home domain. Therefore, AAAL communicates with the MN’s AAAH via e.g. the Diameter protocol;

- For the mobility management in a HMIPv6 architecture, the binding update process must be performed at Mobile Anchor Point (MAP) before the binding udpate operations at MN’s Home Agent (HA) and Corresponding Node (CN);

- A resource reservation protocol should be used to establish a QoS path from MN to its CN; To support Differentiated Services (DiffServ), the related policies should be enforced at the AR;

- To minimize the initial registration latency, the QoS reservation operations and binding update operations should be integrated in one process;

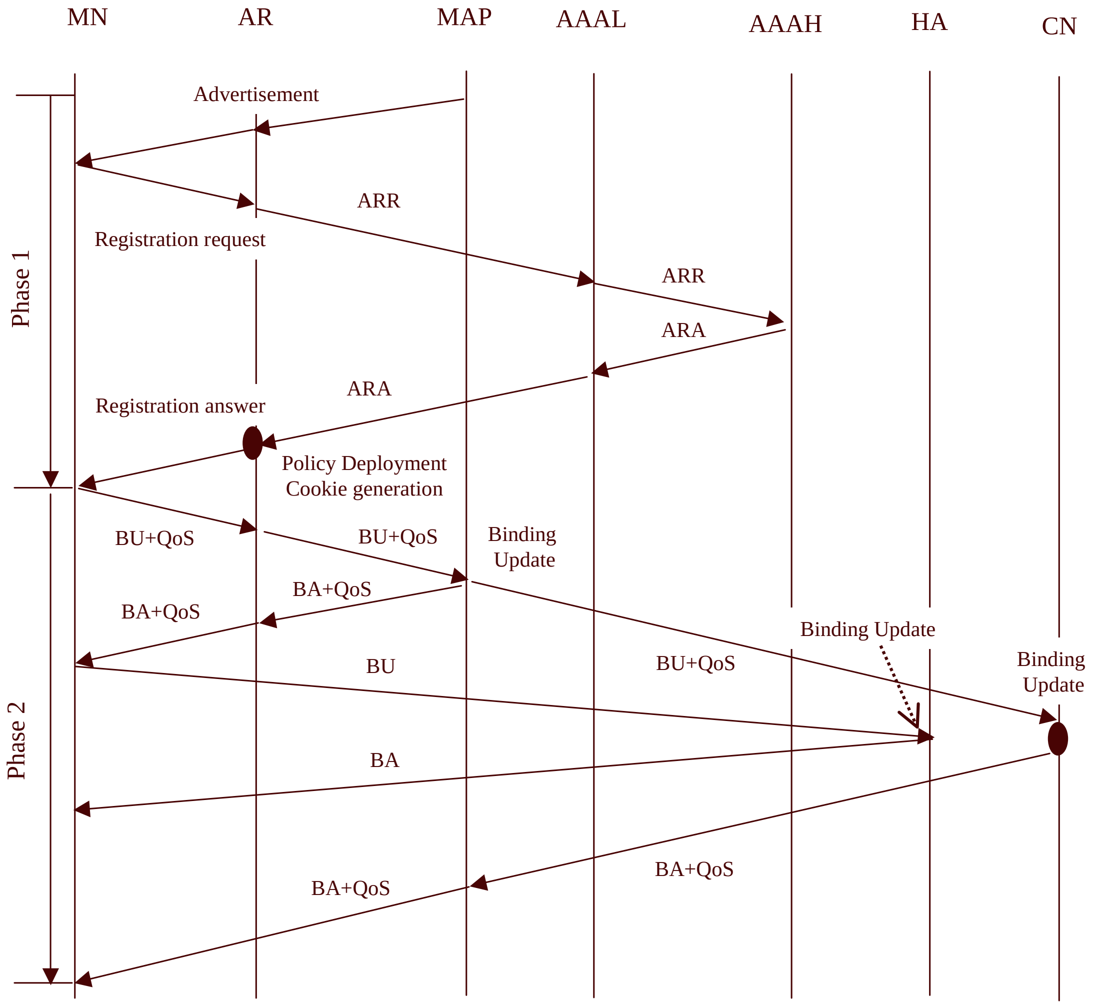


Figure B.1: The Signaling Procedure in the Power-up Cases

- A cookie and other related information should be issued to the MN after a successful initial registration in order to be used in the next intra-domain handover.

An example initial registration procedure is given as Figure B.1.

In Phase 1, an Authentication and Authorization (AA) check is performed. The Diameter protocol can be used. After a successful AA check, the policies to support DiffServ can be deployed at Access Router (AR); also AR generates a cookie and other related information which will be used in the next intra-domain handover (see details in the first step of the cryptographic protocol in the intra-domain handover section).

As results of the successful procedure in Phase 1, the local network obtains the information of the mobile node:

- AAAL caches the authorization data of the mobile node or the Service Level Agreement (SLA) in order to perform subsequent re-authorization requests locally;

- the access network obtains the session key for the request;
- AR sets up a IPSec security association with the MN.

The signaling process for the security check is Phase 1 may be misused by an attacker to launch Denial of Service (DoS) attacks.

In Phase 2, also the signaling can be protected with the security association. In the binding update process at MAP, resource reservation signaling can be integrated. When binding update has been performed successfully at MAP, MAP sends “BA+QoS” message towards MN. Meanwhile, MAP can initiate an joint binding update and resource reservation process towards the Corresponding Node (CN). The corresponding acknowledgement message arrives at MAP, MAP forwards the “BA+QoS” message from CN to MN. When MN receives the “BA+QoS” from MAP, it performs binding update at the home agent.

Some alternative solutions for Phase 2 exist. For example, a.) MN may first perform a binding update operation with MAP without resource reservation; when receiving the acknowledgement message, it starts the binding update process with HA and an integrated BU+QoS towards CN separately; or b.) in order to reduce the traffic over the wireless channel between MN and AR, MAP initiates the BU process with HA on behalf of MN and it holds the BA+QoS message until the corresponding BA+QoS message from CN. The optimization of the QoS signaling in mobility scenarios is still a challenge in the Next Steps in Signaling (NSIS) working group of Internet Engineering Task Force (IETF).

B.2 Protocol Operations of the Inter-Domain Handover Case

The protocol operations in the inter-domain handover cases are similar with those in the power-up cases. However, one optimization can be made:

In the propagation of the BU+QoS message towards CN, the QoS path establishment is complete when the message hits the Cross-over Router (CR), which the the joint node where the old path and the new path meet.

Appendix C

Experimental Environment Installation Manual

This chapter includes the following sections:

- Hardware and Network Topology Setup
- Nodes OS Configuration
- MIPL MIPv6 Setup
- CASP Mobility Client Protocol Setup

C.1 Hardware and Network Topology Setup

Figure C.1 is a typical experimental network topology [67]. It is capable to simulate all relevant scenarios such as local and regional handover. Local handovers are performed by using CASP Mobility Client Protocol depending on resource availability in the visited AN.

All machines are i386 Pentium II 300MHz. The machines network cards are 3Com 100BaseT Network Interface Card (NIC)s. A manageable HUB is used for emulating the movement between the links 0, 6, 7 and 8. Links 1 and 4 are connected each with a simple 10M HUB. The terms i0 and i1 refer to interface 0 (eth0) and interface 1 (eth1).

To configure a virtual link (dummy0 interface) on each MAP according to the example configuration file is given in Table C.4.

To Enable the new dummy0 interface: /etc/init.d/network restart

Configuration files for enabling the virtual link on both MAPs are shown in Table C.5 and C.6

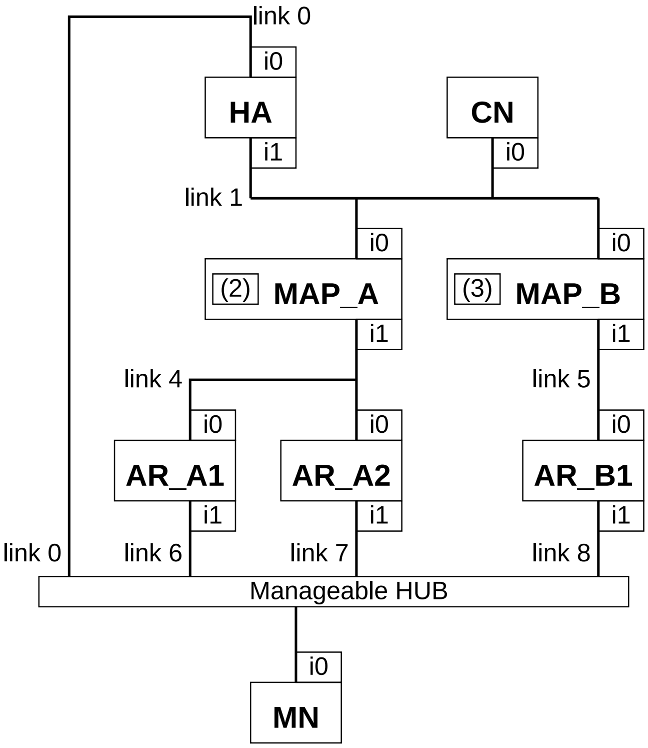


Figure C.1: Example Network Topology Setup

C.2 Node OS Configuration

Redhat 7.2 is used as Linux distribution on all machines. The sources provided with this package only match Linux kernel version 2.4.7. MIPL package mipv6-0.9-v2.4.7. Router advertisement daemon radvd-0.7.1 has been modified to support enhanced advertisement scheme.

C.3 MIPv6 Setup

- HA, CN, AR_A1, AR_A2 and AR_B1and MN are configured according to the example configuration files given in the Tables C.1 (for the HA), C.9 (for the AR), and C.2 (for the MN).

- HA, MAP_A, MAP_B, AR_A1, AR_A2 and AR_B1 have the enhanced radvd installed. Each radvd is configured to advertise the router's availability on the link connected with the manageable HUB. An example configuration file for the enhanced advertisement is shown in Table C.3

C.4 CASP Mobility Protocol Setup

C.4.1 Enhanced Advertisement Support on MAPs and ARs

- To install enhanced radvd-0-7-1, copy all files in radvd-0.7.1 to the radvd-0.7.1 source directory of MAP_A, MAP_B, AR_A1, AR_A2, AR_B1: /usr/src/raddvd_0.7.1/

- Go to each radvd source directory (at MAPs and ARs):

  - ./configure # only necessary if not already run before
    (alternative: "./configure –prefix=/usr/ –sysconfdir=/etc/
    –mandir=/usr/share/man/ –with-pidfile=/var/run/radvd/radvd.pid")

  - make

  - whereis radvd (This tells you where radvd is.)

  - cp ./radvd /usr/sbin/radvd (in most cases, 'whereis' shows you this directory.)

- Configure /etc/radvd.conf in MAP_A and MAP_B for emission of MAP discovery options. An example configuration file for MAP_A is given in Table C.7.

- Configure /etc/radvd.conf in AR_A1 AR_A2 AR_B1 for propagation of MAP discovery options. An example configuration file for AR_A is given in Table C.8.

- (Re)start radvd: "/etc/init.d/radvd restart"
  or directly from shell in the radvd directory by typing: "./radvd -C /etc/radvd.conf -d4 -m stderr". In order to make the MAP options perceptible for the AR before they are advertised by the MAPs, the last step should be performed at the ARs first.

The syntax is the same as the original radvd options syntax described by "man radvd". Examples, how to use the new configuration options, are given in Table C.7 (for MAP) and C.8 (for IR).

C.4.2 Mobility Support on HA, MAPs, ARs and MN

- Copy all files from mobile_ip6 to the mobile_IP6 source directory of the MN (in general /usr/src/linux/net/ipv6/mobile_ip6/).

- Recompile and restart the mobile_ip6 module:

  - cd /usr/src/linux

  - make modules
  - make modules_install
  - /etc/init.d/mobile-ip6 restart

C.4.3 QoS and Security Support on MAPs, ARs and MN

Before compile and load the casp-mob-module, the mobile_ip6 module should be running (list all loaded modules by the command: “lsmod”) .

- cd /usr/src/linux/net/ipv6/mobile_ip6/daemon/
- make
- insmod qos-monitor.o (This loads the qos-monitor module.)
- lsmod (Check that module is loaded.)

The qos-monitor module must be unloaded before the mobile_ip6 module stops.

- rmmod qos-monitor
- /etc/init.d/mobile_ip6 stop

C.4.4 Proc files on MAPs, ARs and MN

proc-file-system is used

- to specify available resources on a router (MAP, IR, AR);
- to monitor reserved resources of a MN;
- to display the cookie information and handover mode on MN.

A number of new entries are created in router’s proc-file-system on MAPs and ARs:

- /proc/sys/net/ipv6/mobility/ir_av_bw
  This entry contains the currently available bandwidth that is controlled by this node. When a MN successfully reserves bandwidth for a flow, this value is reduced respectively. When a reservation is explicitly released or a reservation’s lifetime exceeds it is increased to its former value. Its default values on MAP, ARa1, ARa2 are 1000, 50, 70 respectively.

- /proc/sys/net/ipv6/mobility/ir_res_bw
  This value contains the currently reserved bandwidth that is controlled by this node. It actually represents the node's total bandwidth - its currently available bandwidth.

The currently reserved resources for a flow as well as certain flow identification parameters can be monitored with the following entries.

- /proc/sys/net/ipv6/mobility/ir_fl1_fl
  This entry indicates the flow label of the flow fl1.

- /proc/sys/net/ipv6/mobility/ir_fl1_lt
  This entry indicates the remaining lifetime of the flow fl1.

- /proc/sys/net/ipv6/mobility/ir_fl1_dbw
  This entry indicates the desired bandwidth requirements of the MN as they are negotiated with other QoS entities on the path, up to this node.

- /proc/sys/net/ipv6/mobility/ir_fl1_abw
  This entry indicates the minimum acceptable bandwidth requirements of the MN that it‘s affiliated BU is conditionalized on.

- /proc/sys/net/ipv6/mobility/ir_fl1_rbw
  This entry indicates the node's currently reserved bandwidth for flow fl1.

- /proc/sys/net/ipv6/mobility/cookie_mn_id
  This entry indicates mobile node's ID in a received cookie (only on ARs).

- /proc/sys/net/ipv6/mobility/cookie_ar_id
  This entry indicates the cookie generator's ID (only on ARs).

- /proc/sys/net/ipv6/mobility/cookie_ct
  This entry indicates the creation time of the cookie (only on ARs).

- /proc/sys/net/ipv6/mobility/cookie_rn
  This entry indicates the random number of the cookie (only on ARs).

- /proc/sys/net/ipv6/mobility/cookie_hash
  This entry indicates the fist byte of the hash code of the cookie (only on ARs).

A number of new entries are created in router's proc-file-system on MN:

- /proc/sys/net/ipv6/mobility/mn_fl1_fl
  A MN's QoS request must be specified for specific flow value. This entry can be used to specify the flow value, used as the flow label of the MN's fl1-QoS object. The remaining QoS requirement parameters of that flow are not valid until the mn_fl1_fl entry contains a positive integer value. Only one flow (fl1) per MN is allowed.

- /proc/sys/net/ipv6/mobility/mn_fl1_lt
  Indicates the remaining lifetime in seconds of the previous successfully performed reservation of low fl1.

- /proc/sys/net/ipv6/mobility/mn_fl1_dbw
  Can be used to specify the desired bandwidth for flow fl1.

- /proc/sys/net/ipv6/mobility/mn_fl1_abw
  Can be used to specify the acceptable bandwidth for flow fl1. This value should be smaller or equal to the desired bandwidth, otherwise the mn_fl1_dbw value is increased respectively.

- /proc/sys/net/ipv6/mobility/mn_fl1_rbw
  Can be read to monitor the currently reserved bandwidth of flow fl1.

- /proc/sys/net/ipv6/mobility/cookie_mn_id
  This entry indicates mobile node's ID in the new cookie.

- /proc/sys/net/ipv6/mobility/cookie_ar_id
  This entry indicates the cookie generator's ID.

- /proc/sys/net/ipv6/mobility/cookie_ct
  This entry indicates the creation time of the cookie.

- /proc/sys/net/ipv6/mobility/cookie_rn
  This entry indicates the random number of the cookie.

- /proc/sys/net/ipv6/mobility/cookie_hash
  This entry indicates the fist byte of the hash code of the cookie.

- /proc/sys/net/ipv6/mobility/handover_mode
  This entry indicates the current handover mode (eager or lazy) of MN.

These entries can be read by the cat command (eg ”cat /proc/sys/net/ipv6/mobility/debuglevel”) and modified with the echo command (eg echo 2 > /proc/sys/net/ipv6/mobility/debuglevel).

map_info_loop tool is running on MAPs, IRs, and ARs, mn_info_loop tool is running on MN, in order to monitor all QoS and binding entries.

```
FUNCTIONALITY=ha
DEBUGLEVEL=1
TUNNEL_SITELOCAL=yes
MOBILENODEFILE=/etc/mipv6_acl.conf
```

Table C.1: Example Configuration File for HA:/etc/sysconfig/network-mip6.conf [67]

```
FUNCTIONALITY=mn
DEBUGLEVEL=2
TUNNEL_SITELOCAL=yes
HOMEADDRESS=3ffe:e:e:0::9/64
HOMEAGENT=3ffe:e:e:0::1/64
RTR_SOLICITATION_INTERVAL=1
RTR_SOLICITATION_MAX_SENDTIME=5
```

Table C.2: Example Configuration File for MN:/etc/sysconfig/network-mip6.conf [67]

```
interface eth0
{
AdvSendAdvert on;
MinRtrAdvInterval 1;
MaxRtrAdvInterval 1.5;

AdvHomeAgentFlag off;
AdvIntervalOpt on;

prefix 3ffe:e:e:0::1/64
{
AdvOnLink on;
AdvAutonomous on;
AdvRouterAddr on;
};

};
```

Table C.3: Example Configuration File for HA’s Advertisements
Example Configuration File for HA:/etc/radvd.conf [67]

```
DEVICE=dummy0
BOOTPROTO=static
BROADCAST=192.168.233.255
IPADDR=192.168.233.4
NETMASK=255.255.255.0
NETWORK=192.168.233.0
ONBOOT=yes

IPV6INIT="yes"
IPV6ADDR="3ffe:e:e:2::4/64"
```

Table C.4: Example Configuration File for MAP:/etc/sysconfig/.../ifcfg-dummy0 [67]

```
ALLOW 3ffe:e:e:2::/64
```

Table C.5: Example configuration file for MAP:/etc/mipv6_acl.conf [67]

```
FUNCTIONALITY=ha
DEBUGLEVEL=2
TUNNEL_SITELOCAL=yes
MOBILENODEFILE=/etc/mipv6_acl.conf
```

Table C.6: Example configuration file for MAP:/etc/.../network-mip6.conf [67]

```
interface eth1
{
AdvSendAdvert on;

MinRtrAdvInterval 20;
MaxRtrAdvInterval 30;

AdvHomeAgentFlag off;
AdvIntervalOpt on;

MapOptRecv off;
MapOptProp on;

prefix 3ffe:e:e:4::4/64
{
AdvOnLink on;
AdvPreferredLifetime 200;
AdvValidLifetime 300;
AdvAutonomous on;
AdvRouterAddr on;
};

mapaddr 3ffe:e:e:2::4
{
MapOptAdv on;
MapOptPref 5;
MapOptLifetime 500;
};

};
```

Table C.7: Example Configuration File for MAP:/etc/radvd.conf [67]

```
interface eth0
{
AdvSendAdvert on;
MinRtrAdvInterval 400;
MaxRtrAdvInterval 600;
AdvDefaultLifetime 600;

AdvHomeAgentFlag off;
AdvIntervalOpt on;

MapOptRecv on;

prefix 3ffe:e:e:4::7/128
{
AdvOnLink on;
# AdvValidLifetime 300;
AdvAutonomous on;
# AdvRouterAddr on;
# AdvIntervalOpt on;
};

};
interface eth1
{
AdvSendAdvert on;
MinRtrAdvInterval 1;
MaxRtrAdvInterval 1.5;

AdvHomeAgentFlag off;
AdvIntervalOpt on;

MapOptRecv off;
MapOptProp on;

prefix 3ffe:e:e:6::7/64
{
AdvOnLink on;
AdvPreferredLifetime 2;
AdvValidLifetime 2;
AdvAutonomous on;
AdvRouterAddr on;
};

};
```

Table C.8: Example Configuration File for AR:/etc/radvd.conf [67]

```
FUNCTIONALITY=cn
DEBUGLEVEL=2
TUNNEL_SITELOCAL=yes
```

Table C.9: Example Configuration File for AR:/etc/sysconfig/network-mip6.conf [67]

Appendix D

Acronyms

AA Authentication and Authorization

AAA Authentication, Authorization, Accounting

AAAL Local AAA Server

AAAH Home AAA Server

AABW Aggregate Available Bandwidth

ABW Acceptable Bandwidth

AES Advanced Encryption Standard

AN Access Network

APs Access Points

AR Access Router

ARR AA-Registration Request

ARA AA-Registration Answer

AT Attacker

AVP Attribute Value Pair

BB Bandwidth Broker

BU Binding Update

BA Binding Acknowledgement

CA Certificate Authority

CASP Cross Application Signaling Protocol

CBW Cached Bandwidth Value at AAAL

CDMA Code Division Multiple Access

CMS Cryptographic Message Syntax

CN Corresponding Node

CoA Care-of Address

CR Cross-over Router

CT Context Transfer

DAD Duplicate Address Detection

DBW Desired Bandwidth

DES Data Encryption Standard

DiffServ Differentiated Services

DoS Denial of Service

DSCP DiffServ Code Point

EAP Extensible Authentication Protocol

EP Enforcement Point

FDMA Frequency Division Multiple Access

FTP File Transfer Protocol

FMIPv6 Fast Handovers in Mobile IPv6

GCoA Global Care-of Address

GSM Global System for Mobile Communication

GPRS General Packet Radio Service

GW Gateway

HA Home Agent

HMIPv6 Hierarchical Mobile IPv6

IEEE Institute of Electrical and Electronics Engineers

IETF Internet Engineering Task Force

IKE Internet Key Exchange

Inter Domain HO Inter Domain Handover

Intra Domain HO Intra Domain Handover

IntServ Integrated Services

IPSec IP Security

IR Intermediate Router

ISO International Standardization Organization

ISP Internet Service Provider

L2 layer-2

L3 layer-3

IP Internet Protocol

LAN Local Area Network

LCoA Local Care-of Address

LERS Localized Enhanced-Routing Schemes

KDC Kerberos Distribution Center

MAC Message Authentication Code

MAP Mobile Anchor Point

MDC Modification Detection Code

MSC Mobile service Switching Center

MIPv6 Mobile IPv6

MIT Massachusetts Institute of Technology

MN Mobile Node

ND Neighbor Discovery

NIC Network Interface Card

NSIS Next Steps in Signaling

NTLP NSIS Transport Layer Protocol

NSLP NSIS Signaling Layer Protocol

OSI Open System Interconnection

PAA Proxy Agents Architecture Schemes

PANA Protocol for Carrying Authentication for Network Access

PBW Available Bandwidth

PGP Pretty Good Privacy

PHB Per-Hop-Behaviors

QoS Quality of Service

QoSB QoS Broker

QoSBU QoS-conditionalized Binding Update

RADIUS Remote Access Dialing User Service

RBW Reserved Bandwidth

RSVP Resource ReSerVation Protocol

SA Security Association

SBW Subscribed Bandwidth

SDL Specification and Description Language

SIM Subscriber Identity Module

SLA Service Level Agreement

SNMP Simple Network Management Protocol

SPI Security Parameters Index

TBD To be Determined

TDMA Time Division Multiple Access

TLS Transport Layer Security

TTP Trusted Third Party

UMTS Universal Mobile Telecommunication System

VoIP Voice over IP

VPN Virtual Private Network

WLAN Wireless LAN

WEP Wired Equivalent Privacy

Bibliography

[1] 3DES Source Code. http://www.cr0.net:8040/code/crypto/des/.

[2] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz. Extensible Authentication Protocol (EAP), June 2004. RFC 3748.

[3] F. Alfano, P. McCann, T. Towle, R. Ejzak, and H. Tschofenig. Requirements for a QoS AAA Protocol, October 2003. Internet Draft: draft-alfano-aaa-qosreq-01.txt.

[4] Analysis of Existing Quality of Service Signaling Protocols, June 2003. draft-ietf-nsis-signalling-analysis-02.txt.

[5] W. Arbaugh, N. Shankar, and Y. Wan. Your 802.11 wireless network has no clothes. In *Proceedings of the First International Conference on Wireless LANs and Home Networks (Singapore, 2001)*, 2001.

[6] T. Aura, P. Nikander, and J. Leiwo. DOS-resistant authentication with client puzzles. *Lecture Notes in Computer Science*, 2133:170, 2001.

[7] Y. Bernet, P. Ford, R. Yavather, and F. Baker. A framework for Integrated Services Operation Over diffServ Networks. RFC 2998, November 2000.

[8] Y. Bernet, P. Ford, R. Yavatkar, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, J. Wroclawski, and E. Felstaine. A Framework for Integrated Services Operation over Diffserv Networks, November 2000. RFC 2998.

[9] D. Bernstein. SYN Cookies. The Linux Documentation Project.

[10] D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, 2nd edition, 1991.

[11] U. Black. *Voice Over IP*. ISBN 0-13-065204-0. Prentice Hall, 2002.

[12] S. Blake, D. Black, M. Calrson, E. Davies, Z. Whang, and W. Weiss. An Architecture for Differentiated Services. RFC 2475, December 1998.

[13] R. Bless, X. Fu, R. Hancock, S. Jeong, S. Lee, J. Manner, P. Mendes, and H. Tschofenig. Mobility and Internet Signaling Protocols, January 2004. Internet-Draft: draft-manyfolks-signaling-protocol-mobility-00.txt.

[14] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: an Overview. RFC 1633, June 1994.

[15] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReserVation Protocol (RSVP – version 1 functional sepcification, September 1997. RFC2205.

[16] P. Calhoun, T. Johansson, and C. Perkins. Diameter Mobile IPv4 Application, August 2004. Internet draft, draft-ietf-aaa-diameter-mobileip-20.txt.

[17] T. Chen, S. Hermann, and G. Schaefer. Secure, QoS-enabled Mobility Support in All-IP Networks - Final Report of the SeQoMo/SAM Project. Technical report, TKN TU-Berlin, 2003.

[18] T. Chen, G. Schaefer, C. Fan, S. Adams, M. Sortais, and A. Wolisz. Denial of Service Protection for Optimized and QoS Handover Based on Localized Cookies. In *proceedings of The 5th European Wireless Conference Mobile and Wireless Systems beyond 3G EW2004*, February 2004.

[19] G. Chiruvolu, A. Agrwal, and M. Vandenhoute. Mobility and QoS support for IPv6 Based Real-time Wireless Internet Traffic. In *IEEE Int'l Conf. Commun.*, volume 1, pages 334–338, 1999.

[20] D. Cohen. Issues in Transnet Packetized Voice Communications. In *Proceedings of the Fifth IEEE Data Communications Symposium, Snowbird, UT, USA*, 1977.

[21] B. Davie, A. Charny, J. Bennett, K. Benson, J. Boudec, W. Courtney, S. Davari, V. Firoiu, and D. Stiliadis. An Expedited Forwarding PHB (Per-Hop Behavior, March 2002. RFC 3246.

[22] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification, December 1998. RFC 2460.

[23] R. Koodli (editor). Fast Handovers for Mobile IPv6, October 2004. Internet Draft, draft-ietf-mipshop-fast-mipv6-03.txt.

[24] H. Einsielder and et al. The Moby Dick Project: A Mobile Heterogeneous All-IP Architecture. http://www.ist-mobydick.org.

[25] S. Faccin, B. Patil, and C. Perkins. Diameter Mobile IPv6 Application, April 2003. Internet-Draft: draft-le-aaa-diameter-mobileipv6-03.txt.

[26] A. Festag. *Mobility Support in IP Celluar Networks - A Multicast-Based Approach*. PhD thesis, TKN, TU-Berlin, Germany, 2003.

[27] I. Fodil and V. Ksinant. User Service Management in Hot Spot Networks Using Policies. In *proceedings of The 5th European Wireless Conference Mobile and Wireless Systems beyond 3G EW2004*, 2004.

[28] D. Forsberg, Y. Ohba, B. Patil, H. Tschofenig, and A. Yegin. Protocol for Carrying Authentication for Network Access (PANA), May 2004. Internet-draft, draft-ietf-pana-pana-04.txt.

[29] X. Fu, H. Karl, and C. Kappler. QoS-Conditionalized Handoff for Mobile IPv6. In *Proc. of the Second IFIP-TC6 Networking Conf. - Networking2002*, pages 721–730, Pisa, Italy, May 2002. Springer-Verlag.

[30] X. Fu, P. Mendes, H. Schulzrinne, and H. Tschofenig. Mobility Issues in Next Steps in Signaling, October 2003. Internet Draft: draft-fu-nsis-mobility-01.txt.

[31] L. Gong and P. Syversion. An approach to designing secure protocols. *Dependable Computing for Critical Applications*, 1998.

[32] L-N. Hamer, B. Gage, B. Kosinski, and H. Shieh. Session Authorization Policy Element, April 2003. RFC 3520.

[33] L-N. Hamer, B. Gage, and H. Shieh. Framework for Session Set-up with Media Authorization, April 2003. RFC3521.

[34] D. Harkins and D. Carrel. The Internet Key Exchange (IKE), November 1998. RFC2409.

[35] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. Assured Forwarding PHB Group, June 1999. RFC 2597.

[36] A. Hess and G. Schäfer. Performance Evaluation of AAA / Mobile IP Authentication. In *Proc. of 2nd Polish-German Teletraffic Symposium (PGTS'02)*, Gdansk, Poland, September 2002.

[37] T. Hiller. cdma2000 Wireless Data Requirements for AAA, June 2001. RFC 3141.

[38] R. Hinden and S. Deering. IP Version 6 Addressing Architecture, July 1998. RFC 2373.

[39] Dieter Hogrefe. *Estelle, LOTOS und SDL: Standard-Spezifikationssprachen für verteilte Systeme*. Springer-Verlag, 1989.

[40] R. Housley. Cryptographic Message Syntax (CMS), August 2002. RFC 3369.

[41] R. Housley, W. Ford, W. Polk, and D. Solo. Internet X.509 Public Key Infrastructure Certificate and CRL Profile, jan 1999. RFC 2459.

[42] A. Jamalipour. *The Wireless Mobile Internet Architecture, Protocols and Services*. Wiley, 2003.

[43] D. Johnson, C. Perkins, and J. Arkko. Mobility Support in IPv6, June 2004. RFC 3775.

[44] P. Karn and W. Simpson. The Photuris Session Key Management Protocol, November 1997. Internet draft: draft-simpson-photuris-17.txt.

[45] S. Kent and R. Atkinson. IP Authentication Header, Novemeber 1998. RFC 2402.

[46] S. Kent and R. Atkinson. IP Encapsulating Security Payload (ESP), Novemeber 1998. RFC 2406.

[47] S. Kent and R. Atkinson. Security Architecture for the Internet Protocol, November 1998. RFC2401.

[48] S. Kivisaari. MIPL Technical Specification Tik-76.115. Technical report, Helsinki University of Technology, mar 2000.

[49] L. Kleinrock. *Queueing Systems, Vol. 1: Theory*. Wiley, 1975.

[50] J. Kohl and B. Neuman. The Kerberos Network Authentication Service, September 1993. RFC 1510.

[51] R. Koodli. Fast Handovers for Mobile IPv6, October 2003. Internet Draft, draft-ietf-mobileip-fast-mipv6-08.txt.

[52] R. Koodli and C. Perkins. Fast Handovers and Context Transfers in Mobile Networks. *ACM Computer Communication Review*, Vol. 31(No. 5), October 2001.

[53] H. Krawczyk. SKEME: A Versatile Secure Key Exchange Mechanism for Internet. IEEE Proceedings of the 1996 Symposium on Network and Distributed Systems Security, 1996.

[54] J. Kurose and K. Ross. *Computer netowrking: A top-down approach featuring the Internet*. Addison Wesley, 2001.

[55] D. Levine, I. Akylidiz, and M. Naghshineh. A Resource Estimation and Call Admission Algorithm for Wireless Multimedia Networks Using the Shadow Cluster Concept. *IEEE/ACM Transaction on Networking*, vol.5(No.1), 1997.

[56] R. Lopez, Y. Ohba, M. Parthasarathy, and A. Yegin. Protocol for Carrying Authentication for Network Access (PANA), May 2004. Internet-draft, draft-ietf-pana-framework-00.txt.

[57] J. Loughney, M. Nakhjiri, C. Perkins, and R. Koodli. Context Transfer Protocol, June 2004. Internet Draft, draft-ietf-seamoby-ctp-10.txt.

[58] J. Manner, A. Lopez, A Mihailovic, H. Velayos, E. Hepworth, and Y. Khouaja. Evaluation of Mobility and QoS Interaction. *Computer Networks, Volume 38, Issue 2*, 2002.

[59] D. Maughan, M. Schertler, M. Schneider, and J. Turner. Internet Security Association and Key Management Protocol (ISAKMP), November 1998. RFC 2408.

[60] C. Meadows. A cost-based framework for analysis of denial of service in networks. *Journal of Computer Security*, 2001. 9(1/2):143–164.

[61] S. Miller, B. Neuman, J. Schiller, and J. Saltzer. Section E.2.1: Kerberos Authentication and Authorization System, dec 1987. MIT Project Athena.

[62] A. Mokhtar and et al. Initial Design and Specification of The Moby Dick QoS Architecture. Technical report, Motorola Labs, Paris, France, may 2002. IST-2000-25394 Project Moby Dick D0201.

[63] G. Montenegro. Reverse Tunneling for Mobile IP, revised, January 2001. RFC 3024.

[64] T. Narten, E. Nordmark, and W. Simpson. Neighbor Discovery for IP Version 6 (IPv6), December 1998. RFC 2461.

[65] NBS FIPS PUB 46 National Bureau of Standards. Data Encryption Standard, jan 1977. National Bureau of Standards, U.S. Department of Commerce.

[66] U.S. Department of Commerce National Institute of Standards Technology. Advanced Encryption Standard, nov 2001. Federal Information Processing Standards Publication 197, Washington, DC.

[67] A Neumann. Prototypical Implementation and Experimental Testbed Setup of a QoS-Enabled Mobility Concept Based on HMIPv6, October 2003. Diploma Thesis.

[68] A. Neumann, X. Fu, and H. Karl. Prototype Implementation and Performance Evaluation of a QoS-Conditionalized Handoff Scheme for Mobile IPv6 Networks. In *Proc. IEEE Computer Communications Workshop (CCW)*, Laguna Niguel, CA, October 2003.

[69] K. Nickols, V. Jacobson, and L. Zhang. A Two-Bit Differentiated Services Architecture for the Internet. RFC 2638, July 1999.

[70] D. O'Mahony, M. Peirce, and H. Tewari. *Electronic Payment Systems for E-Commerce*. Artech House, 2001.

[71] H. Orman. The OAKLEY Key Determination Protocol, November 1998. RFC 2412.

[72] S. Pack and Y. Choi. Performance Analysis of Fast Handover in Mobile IPv6 Networks, 2003. Lecture Notes in Computer Science (LNCS), Vol. 2775, pp. 679-691, Springer-Verlag.

[73] S. Pack and Y. Choi. A Study on Performance of Hierarchical Mobile IPv6 in IP-based Cellular Networks. *IEICE Transactions on Communications, Vol. E87-B, No. 3,* March 2004.

[74] M. Parthasarathy. PANA enabling IPsec based Access Control, may 2004. draft-ietf-pana-ipsec-03.txt.

[75] C. Perkins. IP Mobility Support for IPv4, revised, June 2004. draft-ietf-mip4-rfc3344bis-01.txt.

[76] A. Perrig, R. Canetti, B. Briscoe, J. Tygar, and D. Song. TESLA:Multicast Source Authentication Transform, July 2000. Internet Draft.

[77] R. Ramjee, T. La Porta, S. Thuel, and K. Varadhan. IP micro-mobility support using HAWAII. Internet draft draft-ramjee-micro-mobility-hawaii-00.txt, February 1999.

[78] R. Ranjee, J. Kurose, D. Towsley, and H. Schulzrinne. Adaptive Playout Mechanisms for Pakcetized Audio Applications in Wide-Area Networks. In *Proceedings of IEEE Infocom 1994, Toronto, Canada*, 1994.

[79] C. Rensing, Hasan, M. Karsten, and B. Stiller. A Survey on AAA Mechanisms, Protocols, and Architectures and a Policy-based Approach beyond: Ax. Technical report, Computer Engineering and Networks Laboratory TIK, ETH Zürich, Switzerland, 2001. TIK-Report No. 111, May 2001.

[80] P. Robert. *Stochastic Networks and Queues*. Applications of Mathematics, Vol. 52, Springer-Verlag, 2003.

[81] P. Roque and L. Fenneberg. RADVD - Raouter Advertisement Daemon, nov 2001. http://v6web.litech.org/radvd/.

[82] Rusty Russell. Linux netfilter Hacking HOWTO, 2001. http://www.netfilter.org/unreliable-guides/netfilter-hacking-HOWTO/.

[83] G. Schäfer. *Security in Fixed and Wireless Networks*. John Wiley & Sons, Ltd, 2003.

[84] B. Schneier. *Applied Cryptography Second Edition: protocols, algorithms, and source code in C*. John Wiley & Sons, Inc., 1996.

[85] H. Schulzrinne. A Quality-of-Service Resource Allocation Client for CASP, March 2003. Internet Draft.

[86] H. Schulzrinne, H. Tschofenig, X. Fu, and A. McDonald. CASP - Cross-Application Signaling Protocol, Febuary 2003. Internet Draft.

[87] H. Schulzrinne, H. Tschofenig, X. Fu, and A. McDonald. Cross-Application Signaling Protocol, March 2003. Internet Draft draft-schulzrinne-nsis-casp-01.txt.

[88] Z. Shelby, D. Gatzounas, A. Campbell, and C. Wan. Cellular IPv6, November 2000. Internet Draft: draft-shelby-seamoby-cellularipv6-00.

[89] C. Shen. Several Framework Issues Regarding NSIS and Mobility, January 2003. draft-shen-nsis-mobility-fw-00.txt.

[90] S. Shenker, C. Partridge, and R. Guerin. Specification of Guaranteed Quality of Service, September 1997. RFC 2212.

[91] H. Soliman, C. Castelluccia, K. El-Malki, and L. Bellier. Hierarchical Mobile IPv6 mobility management (HMIPv6). Internet-Draft: draft-ietf-mipshop-hmipv6-03.txt, October 2004.

[92] J. Solomon. *Mobile IP The Internet Unplugged*. Prentice Hall, 1998.

[93] W. Stallings. *Cryptography and Network Security: Principles and Practice*. Prentice Hall, second edition edition, 1998. Hardcover, 569 pages.

[94] W. Stevens. *TCP/IP Illustrated, Volume 1*. Addison Welsey, 1994.

[95] A. Talukdar, B. Badrinath, and A. Acharya. MRSVP: A Resource Reservation Protocol for an Integrated Services Network with Mobile Hosts. *Wireless Networks*, 7(1):5–19, 2001.

[96] R. Thayer. IP Security Document Roadmap, November 1998. RFC 2411.

[97] VisioWave. VisioWave Dynamic Coding White Paper.

[98] J. Vollbrecht, P. Calhoun, S. farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, and D. Spence. AAA Authorization Application Examples, August 2000. RFC 2905.

[99] J. Walker. Unsafe at any Key Size: An Analysis of the WEP Encapsulation, October 2000.

[100] J. Wroclawski. Specification of the Controlled-Load Network Element Service, September 1997. RFC 2211.

[101] J. Wroclawski. The Use of RSVP with IETF Integrated Services, September 1997. RFC 2210.

[102] L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP) — Version 1 Functional Specification. RFC 2205, September 1997.

Acknowledgement

I would like to express my sincere gratitude and appreciation to my advisor, Professor Adam Wolisz, for offering me the unique opportunity to work in the research area of telecommunications and networking, for his expert guidance and mentorship, and for his encouragement and support at all levels.

I would also like to thank Professor Paul Mueller for reading my dissertation and giving constructive comments.

I give my special thankfulness to Dr. Guenter Schaefer for his inspiring and encouraging way to guide me to a deeper understanding of my work, and his invaluable comments during the whole work with this dissertation.

I am grateful to Dr. Festag for reading my dissertation and giving me much-appreciated advice. I also give my thanks to Sven Hermann for a fruitful cooperation in the SeQoMo project. I also want to thank Dr. Changpeng Fan for providing constructive comments in the project development. Thanks also to Dr. Xiaoming Fu for a wonderful cooperation in the project. The project was funded by Siemens AG.

I would like to thank Axel Neumann for his generous help in the experimental phase of this research.

I am very grateful to Dr. Michel Sortais for his mentorship on mathematics and for proofreading my mathematical analysis chapters. The co-authorship with Dr. Michel Sortais and Dr. Stefan Adams for papers gave a wonderful memory.

Jamarin Phongcharoen and Andreas Krause produced a useful results in a student project “Simulation of A Denial of Service Protection Scheme for Optimized and QoS-aware Handover in Mobile Communication Networks Based on Localized Cookies”. Thanks to their efforts.

I wish to acknowledge my appreciation to Ms. Petra Hutt for her assistance during my Ph.D. study in the group. I appreciate all my colleagues and friends for their friendship and companionship. I had a wonderful time in my nearly 5 year period in the group.

Last not least, I would like to express my special love and thanks to my wife Jing and my daughter Audrey for their endless love and support. I would have finished my dissertation without their support. I owe a lot to my parents for their great loves and cares.

Tianwei Chen