

**Virtual Private Networks for mobile environments.
Development of protocol for mobile security and
algorithms for location update.**

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

Dissertation

zur Erreichung des akademischen Grades
Doktor-Ingenieur (Dr.-Ing.)
von

Dipl.-Ing. Vesselin Dimitrov Tzvetkov

geboren in Sofia, Bulgarien



Referenten: Prof. Dr. Johannes Buchmann
Prof. Dr.-Ing. Ulrike Meyer

Tag der Einreichung: 15. Dezember 2009
Tag der mündlichen Prüfung: 03. Februar 2010
Hochschulkenziffer: D 17

Darmstadt, Februar 2010

Dedicated to my family

Abstract

The classical networks for broadcast, telephony and data are converging to services on the Next Generation Networks (NGN), which are introduced by all major Service Providers (SP). Major requirements on the future IP network are security and mobility, which are reflection of the Internet's importance and wide use of portable smart devices.

Secure IP mobility is the focus of this thesis, i.e. how the user can move through different access networks whilst maintaining uninterrupted and secure IP communication. In particular, the remote access (corporate access) is the prime task, thus remote clients connect to central gateway, where corporate IP address or LAN segments are assigned. The corporate access requires naturally high level of security to protect against competitors. The security must cover the application data and mobile protocol signalling. This thesis targets an implementable solution for IPv4 and IPv6. It must integrate in the existing Service Provider infrastructure, like tunnelling devices (BRAS), AAA, Load Sharing, High Availability, Firewalls, PKI, monitoring, and administration etc.

The existing approaches, like for example: Mobile IP with IPSec, MOBIKE, Proxy Mobile IP, are presented and analysed at first stage. The existing solutions fall short in many areas like: not considering NAT devices, not compatible to multi-homed hosts, without session tracking protection, problems with anti-spoofing rules performed by Internet Providers etc. A major deficit of all existing solutions is that the network parameters are updated at constant intervals. Neither the frequency of the host movements nor the network properties are considered by the update. This leads to underperformance regarding to the network load and convergence time due to disconnection.

In this thesis, a new protocol family is developed, called Mobile VPN (M-VPN). The M-VPN consists of three sub protocols: Mobile Key Exchange (M-KE), Mobile Secure Encapsulation (M-SE), Mobile Location Update (M-LU).

There are two major parts in this work: (1) engineering development of M-SE and M-KE for mobile IP security, and (2) mathematical algorithms (M-LU) for optimisation of the updates in mobile networks. Both parts build a complete view of the remote corporate access in mobile environments.

The M-KE and M-SE have novel characteristics like mobility during the session negotiation through polling and caching, protection against location tracking through pseudo random header values and overlay dynamic topologies through network resources discovery.

The principal idea in M-LU is to make the update interval proportional to the probability of disconnection. The updates are frequent in the timeframe with a high probability of disconnection and vice versa. The probability density function is built using the history of past changes in the parameters. The classical estimation methods cannot be used in a straightforward way in M-LU, since they require numerical values as result from a measurement. Unfortunately, the update procedure delivers only Boolean values and namely if the IP/UDP parameters have changed.

The developed M-LU protocol creates three novel frameworks representing comprehensive and primitive solutions of the problem, thus stochastic, subjective and analytical. They are based on (1) sequential Monte Carlo in Particle filter, (2) Adaptive Fuzzy controller and (3) extended Kalman filter.

A proof of concept on Mobile Location Update protocol is achieved through simulation on Matlab 7.0. The results show clear outperformance of new methods against the constant interval. The novel framework can also be implemented in various protocols like IPSec, SIP or Mobile IP etc.

Zusammenfassung

Sichere Mobilität in IP Netze ist das Hauptthema dieser Dissertationsarbeit und zwar wie ein Benutzer eine sichere Kommunikation während seiner IP Adresse sich ändert betreiben kann? Der Fokus liegt auf mobilen Firmennetzzugriff (Remote access), da er naturgemäß eine höhere Anforderung an der Sicherheit verlangt. Diese Dissertation beschäftigt sich mit anwendbaren Lösungen für IPv4 und IPv6. Eine anwendbare Lösung muss sich in der existierenden Infrastrukturen und Methoden bei den Service Providern, wie Tunneling Geräte (BRAS), AAA, Firewalls, Lastverteilung, Verfügbarkeit, Management, usw, integrieren.

Die Lösungsansätze, wie zum Beispiel Mobile IP mit IPSec, MOBIKE, Proxy Mobile IP, sind zuerst in der Arbeit analysiert. Diese haben Defizite in mehreren Bereichen, wie zum Beispiel: nicht kompatibel zu NAT und Multi-Homed Hosts, keinen Schutz gegen das Verfolgen der Aufenthaltsorte. Zentraler Defizit ist, dass alle Lösungen die Netzparameter in konstanten Intervallen aktualisieren. Weder der Bewegung des mobilen Hosts noch die Netzparameter sind dabei berücksichtigt. Das führt zu regelmäßigen Verbindungsabbrüchen und/oder Netzüberlastung durch nutzlose Paketen.

In dieser Arbeit wird eine neue Protokollsammlung, genannt Mobile VPN (M-VPN), entwickelt. Das M-VPN teilt sich in drei Unterprotokolle auf: Mobile Key Exchange (M-KE), Mobile Secure Encapsulation (M-SE) und Mobile Location Update (M-LU).

Die Dissertation hat zwei Hauptziele: (1) eine Ingenieuraufgabe zur Protokollentwicklung für mobile IP Sicherheit und zwar M-SE und M-KE. (2) Die zweite Aufgabe (M-LU) ist die mathematische Optimierung von Netzparameteraktualisierung für reduzierten Netzlast und Verbindungsabbrüche. Beide Teile beschreiben alle Aspekten des Firmennetzzugriffs in einer mobilen Umgebung.

Die M-KE und M-SE führen neue Methoden in die mobilen Sicherheit ein und zwar: Durch Polling und Caching wird IP Änderung während einer Sitzungsaushandlung ermöglicht. Pseudozufallswerte im Header sorgen für einen Schutz gegen das Verfolgen der Aufenthaltsorte. Der Aufbau von dynamischen Netztopologien wird durch das Annoncieren von weiteren Mobilern Servern ermöglicht.

Die grundlegende Idee im M-LU besteht darin, die Aktualisierungsfrequenz proportional zu der Wahrscheinlichkeit des Verbindungsabbruchs einzustellen. Wenn es eine höhere Wahrscheinlichkeit für IP Änderung gibt, werden die Netzparameter öfter aktualisiert und vice versa. Die Wahrscheinlichkeitsdichtefunktion wird anhand der Verbindungsabbrüche in der Vergangenheit konstruiert.

Die klassischen Methoden der Signaltheorie können nicht direkt in M-LU verwendet werden, da diese Zahlenwerte einer Messung verlangen. Im Gegensatz liefert die Prozedur für die IP/Port Aktualisierung ein einfaches Booleschen Ergebnis und zwar, ob sich die Netzparameter geändert haben. Die Booleschen Werte können nicht in den klassischen Algorithmen eingesetzt werden. Der Zeitpunkt der Änderung liegt in dem Intervall zwischen zwei Aktualisierungen.

Drei mathematische Algorithmen werden für M-LU entwickelt. Diese decken die grundlegenden Ansätze für eine Lösung und zwar stochastischer, subjektiver und analytischer Ansatz. Diese basieren auf: (1) Sequentielles Monte Carlo mit Particle Filter, (2) Adaptive Fuzzy Kontroller, (3) erweiterten Kalman Filter.

Das M-LU Protokoll wurde mit Matlab 7.0 simuliert, um die Qualität der Methoden zu prüfen. Das neue Verfahren hat eine deutlich bessere Effektivität und Genauigkeit verglichen mit den konstanten Intervallen. Die hier entwickelten Verfahren können in einer Reihe von weiteren Protokollen, wie zum Beispiel SIP, IPSec oder Mobile IP, implementiert werden.

Acknowledgements

I am deeply grateful to all people who contributed to this work. I would like to thank Prof. Johannes Buchmann for his significant support in the final stage of my work. The second referent Prof. Ulrike Meyer helped me to improve the presentation part and to structure the thesis. Especially, her deep knowledge in networks and security led to many improvements.

My research will be impossible without direct encouragement and support by Dr. Volker Sebastian, Dr. Walter Häffner and the Vodafone Management for presenting papers at many conferences. Prof. Aleksander Tsenov and Dr. Tim Wichmann read the mathematical part of my work and gave me important feedback. I would like to thank my colleagues at service development department at Vodafone (former Arcor AG&Co KG) for the creative and inspiring discussions.

I would like to thank my wife and our children for their support during the many years of research in parallel to my work at Vodafone. This work is dedicated to them with all my love. My education and research were motivated by my parents, who encouraged me from the very beginning. Thank you!

December 2009

Vesselin Dimitrov Tzvetkov

Erklärung¹

Hiermit erkläre ich, dass ich die vorliegende Arbeit, abgesehen von den in ihr ausdrücklich genannten Hilfen, selbständig verfasst habe.

Wissenschaftlicher Werdegang des Verfassers in Kurzfassung

1997 - 2001	Studium an der Ruhr-Universität Bochum. Abschluss Dipl.Ing. in ET mit Nebenfachstudium der Informatik. Diplomarbeit: Sicherheit in WAP Protokoll.
2000	Projektarbeit an University of Sheffield (Großbritannien), Sicherheitsprotokolle in mobiler Umgebung.
1993 - 1997	Studium an der TU Sofia (Bulgarien), Fachrichtung Kommunikationstechnik.

¹ gemäß der Promotionsordnung der TU Darmstadt

Contents

1	Introduction.....	1
1.1	Goals	1
1.2	State of the art	2
1.3	Motivation	2
1.4	Structure of the thesis	3
1.5	Properties and ideas of M-VPN.....	4
1.6	Contributions	5
1.7	Background information.....	5
1.8	References in chapter 1	8
2	Secure IP mobility	9
2.1	High-level description of secure IP mobility	9
2.2	Definitions	10
2.3	Influence of network change on the mobility	11
2.4	Influence of NAPT on the mobility	11
2.5	Multi-homed hosts and mobility	13
2.6	Tracing of physical location in mobile environments	14
2.7	Requirements for secure IP mobility	15
2.8	Principles of mobility	17
2.9	Tunneling for enabling mobility.....	18
2.10	Existing approaches for secure IP mobility	19
2.11	Related research projects for secure mobile networks.....	35
2.12	Conclusion.....	37
2.13	References in chapter 2	39
3	Mobile Virtual Private Network	41
3.1	Principles of Mobile VPN	41
3.2	Targets of Mobile VPN	42
3.3	Architecture overview of Mobile VPN	45
3.4	Bundling of M-KE and M-SE	49
3.5	M-KE overview	50
3.6	M-SE overview	52
3.7	Anti-tracing mechanism in M-SE.....	54
3.8	Dynamic server discovery	57
3.9	Security Associations Management Database (SAM-DB).....	57
3.10	Non interrupted operation in M-SE.....	59
3.11	Location update notification in M-SE	59
3.12	Dead peer detection and NAPT keep-alive	59
3.13	TCP in TCP Tunnel.....	60
3.14	Summary and contributions.....	60
3.15	References in chapter 3	62
4	Mobile Key Exchange	63
4.1	Authentication methods.....	63
4.2	M-KE messages.....	64
4.3	Packet structure	66
4.4	ClientResponse.....	70
4.5	ServerResponse	71
4.6	Key derivation	72
4.7	Fragmentation of Mobile KE datagram.....	73
4.8	Payloads	74
4.9	Encrypted payloads	74

4.10	Notification.....	74
4.11	Connection redirection	75
4.12	EAP Authentication properties	76
4.13	Appendix M-KE	77
4.14	References in chapter 4.....	85
5	Mobile Session Encapsulation	87
5.1	Tunnel mode	87
5.2	Native mode.....	89
5.3	Packet structure	89
5.4	Packet processing	91
5.5	Anti replay protection	92
5.6	Unprotected notifications.....	92
5.7	Traffic Flow Confidentiality Padding.....	92
5.8	Notification.....	93
5.9	L2TP over M-SE	94
5.10	GRE over M-SE	95
5.11	References in chapter 5.....	96
6	Security properties of M-VPN.....	97
6.1	Authentication at different layers	97
6.2	Network security policy.....	99
6.3	State attacks	100
6.4	Attacks on M-KE and M-SE.....	102
6.5	Perfect Forward Secrecy.....	106
6.6	Oracle services.....	106
6.7	State diagram.....	106
6.8	Considerations regarding buffer overflows and injection attacks.....	110
6.9	Conclusion.....	110
6.10	References in chapter 6.....	111
7	Mobile Location Update protocol	113
7.1	Challenges of the update procedure.....	114
7.2	Solution methods	114
7.3	Abstraction model and terminology	115
7.4	Targets	117
7.5	Simulation and performance evaluation	117
7.6	Porting of updates algorithm in further protocols.....	121
7.7	References in chapter 7.....	125
8	Mobile Location Update with Sequential Monte Carlo methods.....	127
8.1	Contributions	127
8.2	Theory of Particle filter	127
8.3	Monte Carlo methods	129
8.4	Location Update procedure with Particle Filter.....	137
8.5	Simulation.....	144
8.6	Simulation results	146
8.7	Conclusion and future work.....	154
8.8	References in chapter 8.....	155
9	Mobile Location Update protocol with Adaptive Fuzzy controller	157
9.1	Contributions	157
9.2	Fuzzy Logic Systems.....	157
9.3	Fuzzy controller for the Location Update procedure.....	163
9.4	Simulation of M-LU with adaptive Fuzzy Logic.....	168
9.5	Simulation results	170

9.6	Conclusion and future work	179
9.7	References in chapter 9	180
10	Mobile Location Update protocol based on extended Kalman filter.....	181
10.1	Contributions	181
10.2	Model for extended Kalman Filter	181
10.3	Distribution of the Update Time Points.....	182
10.4	Transformation function.....	183
10.5	Approximated Transformation function.....	185
10.6	Coefficients of the transformation function.....	185
10.7	Extended Kalman filter for M-LU.....	188
10.8	Simulation	194
10.9	Simulation results	195
10.10	Conclusion and future work	203
10.11	References in chapter 10	204
11	Application of M-LU in external protocols.....	205
11.1	Optimisation of Dead-Peer-Detection in IKE using M-LU.....	205
11.2	SIP optimisation through M-LU.....	205
11.3	Binding update message in Mobile IP.....	206
11.4	References in chapter 11	207
12	Conclusion.....	209
12.1	Development of Mobile VPNs	209
12.2	Mobile Location Update protocol	209
12.3	Simulation results	210
12.4	Future work	210
A	Appendix - Internet structure and relevant protocols.....	213
A.1	Internet access	213
A.2	Overview of relevant protocols	215
A.2.1	NAPT Overview.....	215
A.2.2	IP Security (IPSec) and IKE.....	224
A.3	References in the appendix A.....	237

1 Introduction

The Next Generation Networks (NGN) [21] are currently rolled out by all carriers around the world. The classical broadcast, telephony and data networks converge to services over the Internet. The global IP network becomes the dominating medium for modern communications. An important requirement of the new services, like VoIP, messaging, Web services, is the mobile usage, for example on smart devices with WiFi or 3G connections. In parallel to the requirement for mobility, security plays a tremendous role for the new services since the information and intellectual property are decisive for whole business branches. The demand for security and mobility is increasing in the NGN networks.

1.1 Goals

This thesis targets a solution for providing secure IP mobility in the context of remote corporate access. The IP mobility means keeping constant IP address from the application perspective whilst changing to different access networks. Security means protecting the application data¹ and the signalling² of the mobile protocol. The Figure 1.1 shows this primary target scenario.



Figure 1.1: Mobile remote access

The Mobile Host is a device frequently changing the access network and keeping secure connection to the Mobile Gateway. The Mobile Host is part of the corporate network, thus it has single IP or LAN segment of the corporate LAN. The Mobile Host has two IPs: (1) The first one is temporary for communication through Internet to Mobile Gateway. The IP changes frequently depending on the access network. The second IP (or LAN) is used for intranet communication in the corporate network and it is constant. The Mobile Gateway assigns and maintains the corporate parameters and therefore, it has administrative right over the Mobile Host.

Generally speaking, the scenario is an overlay topology of two networks: one public and one private. The overlay network is not restricted only to corporate deployment. It is question of abstraction to higher layer to deploy this model also for applications working on peer-to-peer principle.

There are engineering and mathematical scopes in this work. The engineering part is to design solution implementable in the current IPv4 and future global IPv6 network. The solution must be carrier grade, which means: First, it must be implementable in the current Service Provider (SP) infrastructures, thus we are not starting on green field (protection of investment). Second, it must satisfy provider operation requirements. These conditions can be summarized as: Load Balancing, High Availability, using current AAA implementations, easy integration in existing tunneling devices (like BRAS), using standard cryptographic algorithms, routing protocols integration, fast auto reconfiguration, possibility for existing Firewall integration.

¹ Application data is the messages generated by the applications and not related to secure IP mobility.

² The term "signalling" refers to the packets for maintaining a mobile connection, like negotiation, updates etc.

The mathematical scope of this work is to optimize the signalling with respect to less disconnection because of change of IP address and at the same time to keep the transmitted packets minimal. The signalling in a network with hundreds of thousands of hosts, like the current Service Providers networks, should not to be underestimated and its reduction directly reflects to less energy consumption. The reduction of energy reflects in less operation cost and it is friendly to earth's natural resources. This part is relevant not only in the scope of this thesis, but also to other network problems where the signalling can be reduced, like VoIP (SIP), routing protocols etc.

1.2 State of the art

There are numerous protocols focusing separately on mobility or on security described in 2.10. They can be combined in potential solutions for secure IP mobility. The candidates can be classified in three major groups: (1) In the first group, there are two layer solutions consisting of one protocol for mobility and another for security, for example: IPSec [20] over Mobile IP [17, 18]. The mobile layer delivers constant IP connectivity and the security layer protects the data on top of it. Both protocols act independent from each other. (2) The second group of possible solutions attempts to integrate mobility in current security protocols, like MOBIKE [19]. The original core protocol is kept static¹ and some extensions are added in order to enable mobility. (3) The third group contains research projects, like ENABLE [22] and SMA [23]. They target mobility in Intranet environments or are based on IPv6 networks. This thesis targets mobility and security in IPv4 and IPv6 networks. The solution must not require changes on the Internet's intermediate devices, like routers and switches.

1.3 Motivation

Enabling secure IP mobility is not a trivial task: on the one hand, the Internet (IPv4) was not designed to provide host's mobility. The IP addresses are assigned static to physical access network. Physical movement means change of the access network and therefore the IP address. On the other hand, the existing security protocols, like IPSec (see 1.7), require static network parameters and any change of the IP leads to disconnection.

The motivation for this work is that the current solutions (see 1.2) do not meet the requirements for secure connection in mobile environments and optimisation of the signalling regarding disconnection and network load. In the following paragraph, the reasons are briefly described.

1.3.1 Engineering issues

The Internet structures evolve steadily bringing additional issues to the mobility. There are two assumptions related to mobility made decades ago, which are not met in the current Internet. First, there is no transparent bidirectional IP connection between the host because of the Network Address and Port Translation (NAPT, for terminology see 1.7) [11, 12 and 13]. Second, the hosts are often multi-homed because of multiple IP interfaces, and thus can have more than one IP address (see 2.5).

The broadband access uses NAPT in the Customer Premises Equipment (CPE, see A.1) for enabling multiple hosts to share the same public IP address. The host behind NAPT reaches Internet through the public IP of the CPE and not its local IP address. The result of it is that the host is not aware of its public IP. This has a tremendous impact on the mobility because the mobile host is not notified by change of its public IP on the NAPT router. From

¹ The word "static" means constant IP and port parameters during one session.

an Internet perspective, every change of the public IP is host moving¹. The host moves although he keeps the same local parameter and access network. This was not considered in the existing mobile protocols, like Mobile IP [17]. These protocols are not compatible with NAT and therefore, cannot work in the current Internet.

The similar issue occurs on multi-homed hosts having multiple active IP interfaces, like 3G, WiFi etc. The application is not aware of the outgoing interfaces in dynamic mobile environments (see 2.5). The outgoing interface is decisive for the source IP address and not knowing the outgoing interface means not knowing the source IP.

The research on mobility and security tasks must be carried out together since there are security aspects on the mobility protocol. The signalling of the mobile protocols must be protected and not only the application data. Tracing the physical movements of the host is security relevant and not considered in the specifications (see 2.6). In the secure protocols integrated in the mobility solution, like IPSec [20], the session IDs (SPI) can be easily mapped to the used IPs during the session and therefore, to physical locations of the mobile host.

1.3.2 Mathematical optimization issues

The frequency of the signalling messages in mobile environment is very critical for the packet losses due to an undetected disconnection (host movement) [4, 5]. Practically, there is no research on how the updates frequency of the IP can be optimized with respect to minimum disconnection time and network resources. The classical estimation methods cannot be directly applied since they require numerical values. The update of IP/port delivers Boolean result, thus if the IP/port has changed or not. The time point of disconnection (change) is unknown. It is somewhere between the updates. The Boolean result cannot be used in the classical estimation methods. There is trade off between the updates (resources) and disconnection interval. More updates mean more resource and may reduce the disconnection. The current applications use constant update intervals regardless of the network or host properties. The result of this is wasted resources in unnecessary updates and long disconnection intervals. The situation escalates in mobile environment where frequent changes in network are typical. This can lead to collapse of network because of overloaded links with useless update messages. The effect will be also long disconnection of the mobile devices. The update frequency is a significant topic, which is not considered in the current protocols.

1.4 Structure of the thesis

Chapter 2 presents the requirements on secure IP mobility and the candidates. The focus of this section is an analysis of the properties and opened issues of the existing potential solutions. The requirements for mobility and security in the context of this thesis are not met by the existing solutions and therefore, a new protocol, called Mobile Virtual Private Network (M-VPN), is developed. The M-VPN introduces novel features and covers the opened issues described in chapter 2. Its structure and properties are described in chapter 3. The M-VPN is protocols family consists of three sub protocols, which together deliver secure IP mobility:

- Mobile Key Exchange (M-KE) is a protocol for the negotiation of session parameters in mobile environments. The peers are identified, authenticated and authorized during the negotiation. The protocol delivers crypto

¹ The term “move” refers to change of the IP address and/or port. It must not be directly related to physical relocation.

algorithms, session key, session ID etc, for the following protection protocol M-SE. The protocol is fully specified in chapter 4.

- Mobile Secure Encapsulation (M-SE) protects the communication in mobile environments. The protocol secures the application data and mobile signalling through encryption, authentication, integrity check and replay protection. The M-SE has an anti-tracing mechanism, thus the session cannot be matched to IP addresses. The protocol itself works on an application layer and delivers transparent transport layer. The applications can use the transport layer (TCP/UDP) directly or can implement an IP tunnel. The protocol itself does not have tunnel properties. It integrates existing tunnel protocols, like L2TP or GRE. Chapter 5 gives the technical specification. The security of M-SE and M-KE protocols is analysed in chapter 6.
- Mobile Location Update (M-LU) is a mathematical algorithms description rather than a technical specification. The protocol optimises the update intervals regarding minimal disconnection and network signalling. The structure of the M-LU is presented in chapter 7. Three novel algorithms for M-LU are developed in this thesis using approaches of statistics, analytics and fuzzy logic. These three conceptual methods show different view points at the problem and build together comprehensive and primitive solution:
 - Stochastic solution using Sequential Monte Carlo is the first method presented in chapter 8. The method represents a solution of the Bayesian equation using the statistic of past events. The method is known as Particle filter.
 - Adaptive Fuzzy controller using expert knowledge and training methods is the second method described in chapter 9. Fuzzy logic deals with subjective knowledge equivalent to verbal descriptions with multivalent values. The training of the controller is made through One Pass (OP) method. The rules are optimised with Recursive Least Square (RLS) method.
 - Analytical solution based on extended Kalman filter is the last method developed in chapter 10. The famous Kalman filter [24] solves analytically the Wiener problem [26]. Unfortunately, the Kalman filter can be applied only to linear and Gaussian models. Linear models cannot describe the movement of Mobile Node. A common solution presents the extended Kalman filter (EKF) [25], which basically approximates a non-linear system to a linear one.

A proof of concept of M-LU algorithms is achieved through simulation with real and pseudo random data in chapters 8, 9 and 10. The results are compared to each other and to the constant update intervals. There is a clear outperformance of the new methods regarding the disconnection time using the same resources. Chapter 11 shows that the new algorithm can be used in variety of algorithms, like SIP [15] and Mobile IP [17]. Appendix A gives a technical overview of the Internet access types and relevant Internet protocols

1.5 Properties and ideas of M-VPN

The key idea in M-LU protocols is to set the update time points proportional to the probability of disconnection. The updates are frequent if the probability of connection drop is high and vice versa. The Probability Density Function (PDF) is constructed from the history of IP changes (disconnections). The classical estimation methods cannot be directly applied

as already mentioned. These methods require numerical values of measurement but the update procedure is delivering Boolean ones. M-LU creates new models for the movements of the mobile node, which can be used with the classical estimation methods. Three novel approaches are precisely derived in M-LU for solving the optimisation task: based on Particle filter, adaptive Fuzzy controller and extended Kalman Filter.

The M-SE builds dynamic overlay topologies allowing discovery of new resources and auto-reconfiguration by network failures. The M-SE uses semi random header values for protection against tracing of the host's movement. Furthermore, the dual purpose of IP, thus for identification and transport, is split between the M-SE and network layer. The new protocol is friendly to NAPT and multi-homed environments.

The M-KE is designed to work in fast changing networks, where the processing time at the host may exceed the movement interval. Circular dependencies are the big issue for these networks, where the hosts cannot finish a negotiation even if they communicate physically. For example: the mobile host is changing its IP every 2 seconds and the server needs 10 Seconds for the generation of the response message. The server cannot answer in time to the right client's address. M-VPN introduces polling for solving these circular dependencies.

The M-KE and M-SE are innovative because of their structure for working in mobile environment. They rely on well-established and common cryptographic algorithms, which are independent from the secure mobility topic and out of scope of this thesis. The synergy through using common cryptographic algorithms in the M-VPN ensures easy and fast deployment with minimum resources.

1.6 Contributions

This thesis contains many scientific contributions by the author, which are enumerated here in the order of occurrence in the text. In the first place, it is the analysis of the existing potential solutions for secure IP mobility regarding the influence of NAPT and multi-homed hosts. In the second place, a novel M-VPN protocol is created fulfilling the NGN requirements. The secure protocol has new features for building dynamic overlay networks (see 3.3.3), anti-tracing mechanism for the host's movements (see 3.7), polling procedure for negotiation in fast changing networks (see 3.5), the Mappers (see 3.6.1) enable physical separation of mobility and tunnel node.

An important contribution to protocols working in mobile environment and not limited to M-VPN is the creation of M-LU. The protocol minimises disconnection time through optimisation of the update frequency based on the history of host's movements. There are many innovations in M-LU overviewed separately for Particle filter in 8.1, for Fuzzy controller in 9.1 and for extended Kalman filter in 10.1. A practical contribution is the simulation showing the qualities of the new methods.

The author has published papers directly related to this work. The method based on Fuzzy logic has been published by IEEE [4, 5]. The method on Particle and Kalman filter has been published [8, 9] and presented to the IEEE conferences. There is author's IETF draft on Mobile VPNs using Mobile IP [1]. The author has published further papers not related to mobile security in areas of PKI [3, 6], WAP [2], VoIP [7] and security benchmarking [10].

1.7 Background information

This chapter gives general information on Network Address and Port Translation (NAPT) since it is required for understanding this work. More details can be found in A.2.

There is no consensus about the exact terminology, so clarification of the notation shall be provided first. The Network Address and Port Translation (NAPT) is synonym for other popular notations such as NAT (Network Address Translation), PAT (Port Address

Translation), Masquerading, Port forwarding, dynamic NAT or static NAT. The NAPT is a more general term and covers all facets of translation, thus with or without port translation and static or dynamic. It is used in this thesis.

NAPT [11, 12, 13] is a widespread technology for connecting different IP realms, which are not directly routable. Through the translation of the IP/UDP header, both realms are able to establish indirect IP connection. The common implementation is the translation of private IP addresses used in LAN to public IP routable on the Internet. Translations of public IP to public IP or private IP to private IP are also possible.

The NAPT device intercepts the IP packets and replaces the header and payload values, so that the IP packet can be forwarded to/from certain host. The common case is the dynamic translation private IP to public IP in the common DSL Router access scenario. The example in Figure 1.2, the NAPT device intercepts the packets from A to B and translates the private IP 10.0.0.1 in public IP 82.0.0.1. The new IP (82.0.0.1) address must be routable to the NAPT device from the Internet. The host B has public IP, thus 192.1.1.1.

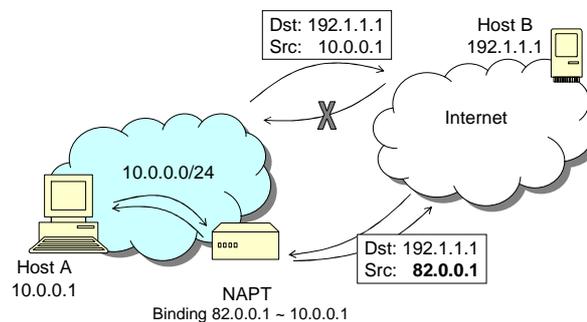


Figure 1.2: An example of NAPT

A NAPT table entry, called *mapping* or *binding*, is created for every session. The mapping (binding) is protocols dependent and can contain the IP and port parameters. The information in the entry (binding) must be sufficient for de-multiplexing the received packets to the right session. The combination of all parameters in the mapping entry must result in a unique session. Otherwise, the packet coming from the Internet cannot be mapped to the private IP.

The NAPT is implemented in all Internet capable devices and operating systems: MS Windows XP/2K, Linux (ip tables), DSL routers and firewalls. The majority of the Internet access is using NAPT, like broadband, WLAN Hotspots, Cable Internet etc. For this reason, it is significant for this thesis.

1.7.1 The controversy of NAPT

NAPT is a very controversial function. It helps the network administrators and ISPs (Internet Service Providers) to grow their networks without changing the routing tables and without providing more IP addresses. This keeps the resources and investments very low. Furthermore, the end customers are no longer restricted to having just one computer, since the ISP assigns just one IP. Most DSL customers have multiple devices, like laptops, PCs, PDAs etc.

NAPT brings enormous restrictions to IP communication. The end host is unaware of its communication address. Many IETF standards are drifted away from the reality, since the assumed transparent bi-directional connectivity is not present. Some of the protocols, like FTP, SIP, H323 etc, were redesigned. There are many documents written about NAPT properties [11, 12, 13, 14]. The major influences of NAPT can be summarised as follows:

- The translation at the NAPT device is unknown to the participants. The hosts are unaware of their translated IP/port. The IP/port changes without any notification.
- Bundled sessions require payload translation, which is not always supported by the devices.
- Dynamical NAPT does not support applications where the initiator is outside. Peer-to-peer services, like VoIP, cannot function in an NAPT environment.
- The P2P applications are forced to deploy different architectures with external proxy. This leads to some further centralisation of services and leaving the P2P principles at network layer.
- The behaviour of NAPT varies dramatically from one implementation to another. It is very difficult for the applications to predict or expose the precise behaviour of NAPT, which may exist in between.
- Robust security in IP environments typically operates on an end-to-end model, where both ends include additional information in the packet to detect manipulation of the packets. NAPT changes the IP and TCP/UDP header values. If the security protocol protects against manipulation of the IP and the TCP/UDP headers, then the NAPT device translation will be treated as an attack. It is not possible to use protection of IP and TCP/UDP in an NAPT environment.
- NAPT has no inherent failover. NAPT is an active in-band mechanism that cannot fail into a safe operating fallback mode. When a NAPT goes offline, all traffic through the device is dropped. An NAPT device is a single point of failure.
- NAPT sits on the data path and attempt to process every packet. Obviously, there are issues regarding the bandwidth scaling.
- With NAPT there is no clear, coherent, and stable concept of network identity. From outside, these NAPT-filtered interior devices are visible only as transient entities.
- Policy-based mechanisms based on network identity, like Policy Quality of Service (QoS), cannot work through NAPT.
- NAPT may drop IP fragments in either direction: without complete TCP/UDP headers, the NAPT may not have sufficient stored states to undertake the correct header translation.

The major influence of NAPT in the Internet cannot be denied or underestimated. Currently, the NAPT feature is implemented in all customer routers (CPEs) and firewalls. Because of the slow migration to IPv6, it can be expected that NAPT will be present further in next decade. The Internet protocols and network architectures must be NAPT friendly in order to work properly.

1.8 References in chapter 1

- [1] Tzvetkov, Vesselin, Sanchez, Erica, "Mobile Virtual Private Network", IETF INTERNET-DRAFT Sheffield University, September 2000.
- [2] Tzvetkov, Vesselin, Cubaleska, Bilijana, "WAP Protocol Security Solutions for Mobile Commerce", 6th Int. Conf. Systemics, Cybernetics, and Informatics (SCI), 2002
- [3] Tzvetkov, Vesselin, "Disaster coverable PKI model based on Majority Trust principle", IEEE ITCC, 2004
- [4] Tzvetkov, Vesselin, "Optimization of update intervals in Dead-Peer-Detection using adaptive Fuzzy Logic", IEEE AINA, 2007
- [5] Tzvetkov, Vesselin, "Fast detection of disconnection using adaptive Fuzzy Logic", IEEE Networking Sensing and Control -ICNSC, 2007
- [6] Tzvetkov, Vesselin, "Decentralization of the Current PKI Infrastructure without Losing Backward Compatibility", IEEE CCNC 4th, 2007
- [7] Tzvetkov, Vesselin, Zuleger, Holger, "Service Provider Implementation of SIP Regarding Security", IEEE AINA Workshops, 2007
- [8] Tzvetkov, Vesselin, "Optimization of mobile updates using Particle filter", IEEE ChinaCom, August 2008
- [9] Tzvetkov, Vesselin, "SIP registration optimization in mobile environments using extended Kalman filter", IEEE ChinaCom, August 2008
- [10] Tzvetkov, Vesselin, "Security level quantification and benchmarking in complex networks", pending IEEE, 2010

- [11] Egevang, K., and P. Francis, "The IP Network Address Translator (NAT)," RFC 1631, May 1994.
- [12] Srisuresh, P., and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations," RFC 2663, August 1999.
- [13] Tsirtsis, G., and P. Srisuresh, "Network Address Translation—Protocol Translation (NAT-PT)," RFC 2776, February 2000.
- [14] Daigle, L., and IAB, "IAB Considerations for Unilateral Self-Address Fixing (UNSAF) Across Network Address Translation," RFC 3424, November 2002.
- [15] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.
- [16] Kadlecsek, József, Pásztor, György, "Netfilter Performance Testing", www.netfilter.org, 2004
- [17] Perkins, C., Ed., "IP Mobility Support for IPv4", RFC 3344, August 2002
- [18] Deering, S., R. Hinden, Editors, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, 1998
- [19] Kivinen, Tschofenig, "Design of the MOBIKE Protocol", RFC 4621, August 2006
- [20] Kent, S., and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005
- [21] ITU-T Focus Group on Next Generation Networks (FGNGN), "NGN 2004 Project description", Version 3, 2004
- [22] ENABLE Project, www.ist-enable.org
- [23] The Open Group, "Secure Mobile Architecture (SMA) Vision and Architecture", Technical Study, February 2004
- [24] Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems", ASME, 1960
- [25] Gelb, A., "Applied Optimal Estimation", MIT Press, 1974
- [26] Wiener, N., "The Extrapolation, Interpolation and Smoothing of Stationary Time Series," John Wiley & Sons, Inc., New York, N.Y., 1949.

2 Secure IP mobility

This chapter focuses on the definition of secure IP mobility and analysis of the existing solutions. A detailed description of the Internet structures and relevant protocols can be found in appendix A, which should be common to the readers. The ideal high uses requirements are defined in 2.1. They are broken down to technical specification in 2.7 considering the physical limitations. The definitions are presented in 2.2. The influence of access network is described in 2.3. The influence of NAT and of multi-homed hosts on the mobility is discussed in 2.4 and 2.5. The principles of mobility protocols are presented in 2.8 and 2.9. The potential solutions are presented and analysed in 2.10. Research projects are discussed in 2.11.

2.1 High-level description of secure IP mobility

In order to communicate, the application opens a network socket, which is common name for bidirectional communication flow. For typical applications, it can be TCP stream socket or UDP datagram socket. From network perspective, the socket is characterized by four major parameters (quadruple): source IP, destination IP address, source port, and destination port. Every change of these four parameters leads to the fact that the application cannot communicate. On the one hand, if the destination IP or port changes, the sent packets get lost, since the values are incorrect. On the other hand, if the application's source IP or port changes, it cannot receive packets. By every change of the quadruple, the socket must be reinitialised with the correct values. For example: if SIP [2] client changes its IP or listener port (UDP), no one can call this client until the SIP registrar is informed of the new IP and port. The notification for the new IP and port is done by registration in SIP [2].

In the context of this work, IP Mobility means keeping the same source IP and ports whilst the host changes the networks. In this way, the application stays reachable and can communicate uninterrupted. The application does not have to re-establish the connection. In other words, Mobile IP is the ability to keep the network and transport layer constant from an application perspective when the host changes its network and transport parameter. The high-level user requirement is: independent of the physical access medium and network to keep uninterrupted application communication.

To achieve uninterrupted communication, the mobile host has virtual and physical network/transport parameters. The virtual parameters stay unchanged and they are used by the application. The physical parameters may change according to the current physical network.

Security of IP Mobility means protection of all exchanged data between the mobile host and its mobile Gateway. This includes the exchanged application data and the signalling by the mobile protocol. The potential attacker is between the hosts, thus somewhere in the Internet. Any information helping an attacker to find private information must be secured. The communication must be protected between the originator and destination host. Solution with protection in hop-by-hop manner, like in SIP [2], is considered insecure. The SIP proxies (the hop) between the SIP clients have access to communication the in clear (without encryption).

The definition of secure IP Mobility leads directly to the primary target scenario – overlay private network. The term “Mobile Virtual Private Network” (Mobile VPN) points out that there is protected overlay network. It can be compared to remote VPN access used in the most corporations. This remote secure access must work in mobile environment, thus mobile VPN. Figure 2.1 presents the scenario. There are three mobile hosts (A, B and C)

connected to mobile gateway. The mobile hosts and mobile gateway participate in the same private network, called corporate LAN in Figure 2.1. All hosts are connected to Internet, thus they are multi-homed having Internet and Corporate network. The Gateway has constant network parameter and the Mobile Hosts change frequently the networks. There are two overlay networks: one public (Internet) and one private (Corporate LAN). As already mentioned in 1.1, the overlay network allows the use of any form of communication, like peer-to-peer, when abstracted to application layer. In fact most of the know peer-to-peer applications, like Skype and Torrent-Clients, have client-server relation at network layer. The users have the same rights administratively, which forms peer-to-peer relation at application

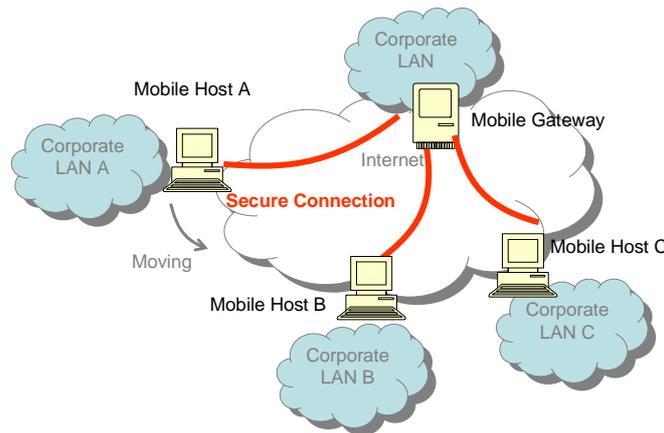


Figure 2.1: Mobile VPN

layer.

2.2 Definitions

Point of Attachment to Internet (PoA): An application is reachable from the Internet at certain IP address and TCP/UDP port. These parameters are defined as the Point of Attachment to Internet. They are a basic requirement to open a socket for the communication with the application. PoA defines an abstraction layer summarising all necessary parameters for communication with one application, such as TCP, UDP, ESP etc. A packet contains two PoAs of the destination and the source application.

Location is synonym for Point of Attachment to Internet (PoA) of the host.

Movement is a change of the PoA. The application is reachable under new IP and port after the movement. The word movement is used in an Internet sense and it may but does not have to be related to physical movement.

Update is a procedure for informing the participant of the new PoA. For example, a registration of Mobile Node to Home Agent in Mobile IP [4]. It consists typically of request and response messages, which must be sent proactively by the hosts, which PoA has changed. An abstraction of the procedure is explained in details in 7.3.

Location Update means update of the PoA parameter and it is a synonym for *update*.

Signalling refers to the messages exchanged by the solution for maintaining the connection, like negotiation, updates etc. They do not contain the application data.

Application data is generated by the end applications, like web, mail etc.

2.3 Influence of network change on the mobility

The Internet address structure defines that static IP ranges are allocated to the access network, thus changing the access network reflects in a change of the IP address. The socket parameters change (PoA) and the applications must establish a new connection. This means by every change of the network is expected a change of the host's IP, thus disconnection. For example: changing (moving) from Wlan Hotspot to 3G (UMTS) will change the host's IP. If host is changing (moving) between access networks in one provider, the host can be assigned the same IP. For example, the 3G (UMTS) network of one operator is geographically huge. The operator can implement solution, like link layer mobility, delivering the same IP in large area. The possibility is limited within the administrative domain of a certain operator and it is physically limited. This is not considered further in this text.

Disconnection and reconnection (up/down event) to same access network does not automatically mean keeping the previous IP address. The Internet Service Provider (ISP) assign the IPs randomly from IP pools. For example, disconnection of DSL line and reconnection will typically change the IP address.

By every re-connection to an access network, an IP address change has to be expected, see Figure 2.2. The host must notify the participants that its IP has changed. Only after successfully update, it can receive packets.

Movement of the host is expected by every change of the access network or reconnection. The mobile application must monitor the interface status and by up/down event must check the IP addresses. The participants must be immediately notified if the IP, and thus the PoA, has changed.

2.4 Influence of NAT on the mobility

The Network Address and Port Translation (NAPT) has a tremendous influence on the Internet connectivity. A NAPT is de facto standard in the broadband access, like DSL. The NAPT properties are presented in A.2.1. Here are discussed only the restrictions related to mobility.

In the transparent Internet without NAPT, movement means local change of the network parameters. It happens mostly because of a change of the access network, which can be related to physical movement. Figure 2.2 shows an abstract example, where the host is changing its local public IP from 1.1.1.1 to 2.2.2.2 and to 3.3.3.3

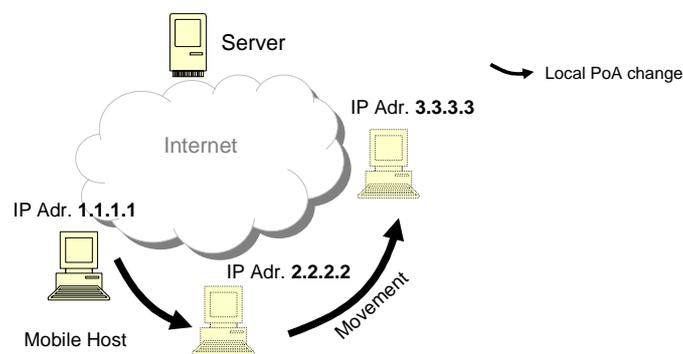
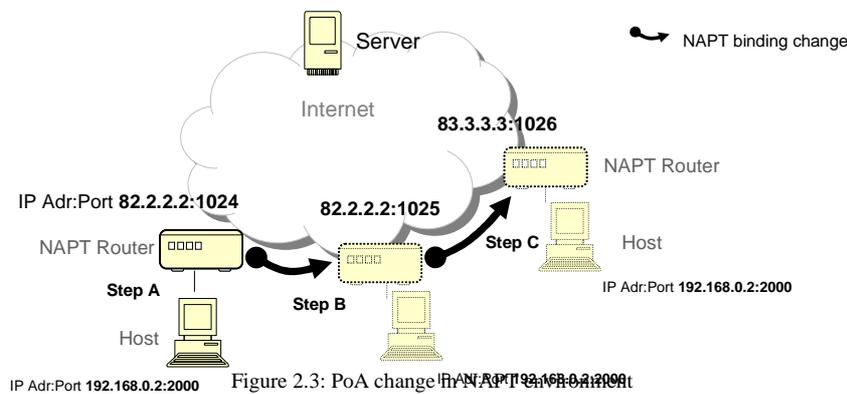


Figure 2.2: Mobile host movement in a Internet without NAT

The situation becomes far more complicated in NAPT environments. When the host is behind a NAPT device, it has private IP address routable only in the LAN behind the router and not in the Internet (see A.2.1). The Internet partners sent packets to the public IP of the NAPT router. The router translates them to private IP of the host in the LAN and forwards

the packets. This means that the PoA from Internet perspective is the one of the NAPT router and not the local network parameters of the host behind the NAPT. The PoA is under control of the NAPT router (NAPT binding entry, A.2.1) and the host behind NAPT is even not aware of it. A change of the binding entry at the NAPT router reflects in new public IP/port parameters of the application. From an Internet perspective, the application is reachable at new port or IP address. The host moves but it is unaware of it since there is no local change of its network parameter. A change of the NAPT can happen due to idle timeout of table entry or due to a new public IP address of the router as described in A.2.1.2.1. For example, the public IP addresses of DSL customers are rotated typically every 24 hours by the Internet Service Provides (ISPs).

The NAPT influence is demonstrated in Figure 2.3 in context of a DSL Router with NAPT and host wired via Ethernet cable to the router. The host has local IP 192.168.0.2 and the application uses port 2000 at the host. The application starts communication to certain Internet host. At step A, the NAPT router creates a binding translation of 192.168.02:2000 to 82.2.2.2:1024. The application is reachable from Internet Server at 82.2.2.2:1024. Assume that the application does not exchange information for e.g. 30 min, which is more than the typical binding idle timeout of 5 min. Then the binding is deleted at the NAPT router. At Step B, the application sends again some information to the same server. A new NAPT binding entry is created. It could be 192.168.02:2000 to 82.2.2.2:1025. The NAPT Router typically uses the next free port in the table. The application is reachable from Internet Server at 82.2.2.2:1025. The PoA has changed, thus the host has moved. At Step C, the application keeps communicating, but the Internet connection is reset by the ISP. The NAPT router is assigned a new public IP address and this causes deletion of all bindings inevitable. The NAPT router creates a new binding on demand using the new IP address, in the example this is binding of 192.168.02:2000 to 82.3.3.3:1026. Again, the PoA has changed without changing the local IP parameters. The host is not aware of the NAPT binding and any change of NAPT device. The host moves even though it does not change its local parameters.



There are two possible reasons for host's movement (PoA change): NAPT table change or reattachment to new access network. The two mechanisms overlay and lead to complex movements of the host. A practical scenario is shown in Figure 2.4. The host is initially attached to the Internet via NAPT router, step A. The NAPT binding is updated and the host obtains new PoA at Step B. Later on at step C, the host, changes the access network and attaches to a new access network with new NAPT device. The mobile host is assigned the same private address as in first step, but the NAPT binding is different (The private address can overlap between different independent LANs). At step D, the host attaches to the Internet without NAPT devices. There is a movement in all of these cases, but local parameters change only at steps D.

Every NATP device between the communication participants adds a degree of mobility. There is additional probability for movement by every additional NATP device. Multiple sequential NATP devices are possible although currently unpopular.

The main problem is that the host is not notified for any binding change at NATP router as described in A.2.1. The host moves on the Internet sometimes without its knowledge as in Figure 2.3. The only way to determine its PoA and consequently a PoA change is to check pro-actively in regular intervals involving update procedure. The host sends notify to partner host which replies with the PoA of initiator in the payload (see definition in 2.2). To recapitulate:

The host is not always aware of its movements regarding PoA, since some changes of PoA are caused by intermediate NATP routers. The NATP router gives an additional degree of mobility in general. The NATP is a significant part of modern Internet access (A.1) and consequently, it must be assumed that the current hosts do not know their PoAs.

In static environments, the applications assume that the NATP binding will not change during the session. This strategy will not work in mobile environments where the network changes are frequent.

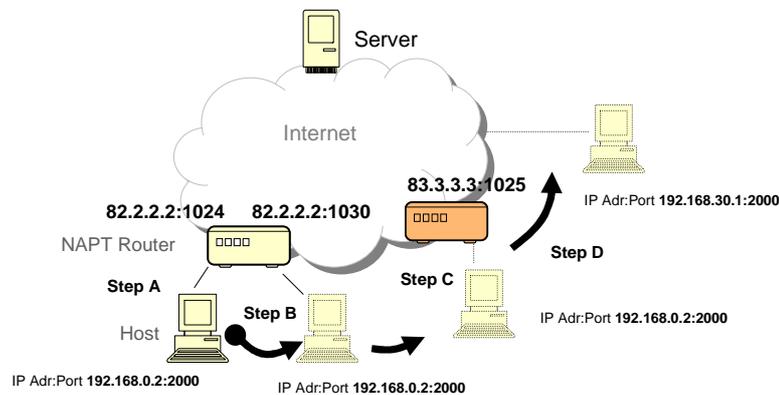


Figure 2.4: PoA change example

2.5 Multi-homed hosts and mobility

Multi-homed host has more than one IP address simultaneously. It is an effect of having multiple interfaces connected to different or the same networks. An example is provided in Figure 2.5. The host has two interfaces, A and B, with two addresses 82.2.2.2 and 145.253.2.2. In the same principle, the host can have a single physical interface, but multiple logical interfaces. Furthermore, the remote corporate access through the Internet causes multi-homed hosts too. The host has one IP address from the corporate network and one public IP assigned by the ISP.

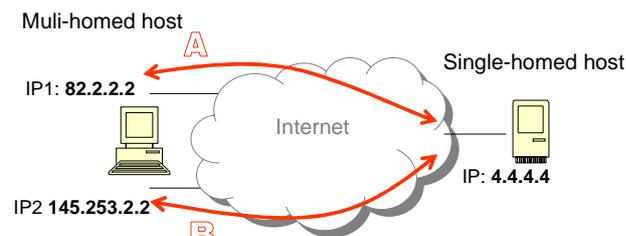


Figure 2.5: Multi-homed host

Most modern smart phones have multiple physical interfaces like GSM, 3G (UMTS), Bluetooth and WiFi. When two or more interfaces are IP connected, the host automatically becomes a multi-homed host. The host has an IP address assigned from WiFi, GSM or UMTS interface.

To understand the difficulties of mobile host with multi-homed hosts, it must be considered the process of setting source IP address at a host. Here is described the decision procedure of the connectionless UDP protocol, since the mobile protocols always operate on UDP, see 3.13. The TCP mechanism is different and out of scope.

The first possibility is that the application can define the source IP address. The application must be pre-configured. Predefining the source IP address causes the use of the same IP address regardless of the outgoing interface. At first sign, it is a good choice. Unfortunately, this is a problem, since the chosen source IP address might not be routable through all interfaces. The Internet Service Providers (ISPs) implement anti-spoofing protection [3]. The anti-spoofing limits the use of source IP to the one assigned to the interface. Different IPs are filtered by the ISPs. Otherwise spoofing attacks will be possible on the Internet. To recapitulate, using a constant IP regardless of the interface is not possible.

The second possibility for choosing the source IP is using the outgoing interface. This is the most common way. The application does not set the source IP and leaves the decision on the routing engine. The routing engine operates at network layer, thus lower than the application. The routing table defines the next hop and the outgoing interface. The IP of the outgoing interface is set as source address of the packet. (The IP addresses are assigned to every interface.)

The routing table can change during one communication session. Those changes could reflect in a change in the outgoing interface and consequently the source IP. The routing table changes are triggered by different events, such as the status of the interface, DHCP configuration etc. If the host is running a dynamic routing protocol, the routing can change due to the update of the table by the remote peer. In order to know the source IP address for sure, the application must monitor the routing table. This is practically impossible since there could be numerous changes.

In static environments, the applications assume that the routing table will not change during the communication. The applications check the routing table at the start up and keep the IP constant. Some applications monitor the state of the interface, which is obviously insufficient in gaining a conclusion on the outgoing interface. This mechanism works in static environments only.

The routing table of multi-homed hosts is decisive for the source IP. It can change independently from the interface's link status. Changes on the routing table are frequent in mobile environments. The application layer is not notified of these changes. Monitoring of the routing table is for the application almost impossible. Consequently, the application is not aware of its PoA changes.

2.6 Tracing of physical location in mobile environments

The mobile and security protocols, like Mobile IP [4, 6] and IPSec [27], have a major issue allowing an intermediate attacker to trace the physical locations of the mobile host. The temporarily IP addresses used by the mobile host for communication can be matched to physical location. Practically, the host movements, like going to hotel, to work, driving on the highway etc. The user ID, like Peter Schmidt, cannot be directly uncovered. The attack gives the movements of certain host. The user ID can be uncovered by using simple heuristics. For example, who is entering the WiFi HotSpot and which session gets active at the same time?

This problem in mobility and security protocols allows making profiling of physical movements, which is a major privacy thread.

The scenario is shown in Figure 2.6 regarding Mobile IP and IPSec. An attacker taps (intercepts) all packets going to Home Agent (HA) or to IPSec Gateway regardless of the IP of the mobile host. This means it must be in front of the IPSec Gateway (HA) in the best case.

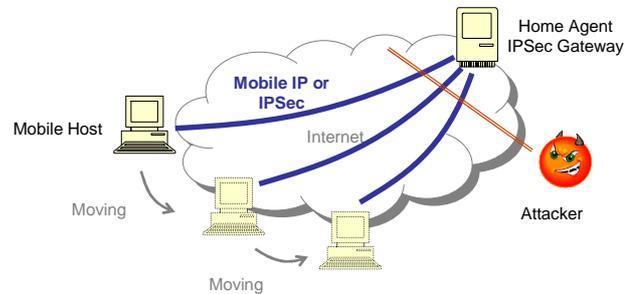


Figure 2.6: Tracking of mobile hosts location

In IPSec, there is data confidentiality but not for the Session ID. The headers of ESP [14] and AH [12] contain the Security Parameter Index (SPI), see A.2.2. The SPI value is unique for the gateway and constant during the IPSec session. The SPI is used to identify the Security Association (SA) for inbound processing at the peer. The SPI value stays constant for a long time and its change can be tracked during the Rekeying in IKE as the NAPT devices does, see A.2.2.10.6 and A.2.1.11. The attacker can match the SPI to all the temporary IPs. The Home Address or User ID is unknown. The attacker has a SPI and all physical locations in the time, thus a profile. As mentioned earlier, the user ID to SPI can be matched using heuristic observation.

In the Mobile IPv4 [4] protocol, the registration message is authenticated but is not confidential. The Home Address of the Mobile Node is sent in clear as part of the registration request. The attacker can easily match the Care-of-Address (temporary IP address) to the Home Address. The attacker can find out where the mobile node is located and how it is moving.

The Mobile IPv6 [6] uses IPSec for protection of the Binding Update which makes the protocols vulnerable in the same way.

The tracking of the mobile session is serious deficit of IPSec and Mobile IP, which are the major protocols for mobility and security. All current solutions, see 2.10, have this issue and it is considered by the author as very serious problem.

2.7 Requirements for secure IP mobility

The high-end requirements defined in 2.1 are described here in technical terms. The common criteria are divided into three groups:

General requirements:

- The mobility must not involve special features by the intermediate devices, such as router, switches etc. The requested functionality must be implemented at the mobile hosts and at the gateway only. The solution must be possible in the heterogeneous Internet, where the devices are under the control of different ISPs.
- The solution must be compatible to the current Internet with NAPT router and multi-homed hosts (see 2.4 and 2.5). The solution should not require a translation of payload by the NAPT routers (see A.2.1).v

- The solution must be supportable in IPv4 and IPv6.
- The solution should integrate in the current network structures, such as AAA (Authentication Authorisation and Accounting) servers, firewalls routers. Without this integration, a practical implementation becomes almost impossible.
- The Internet hosts must not support any features in order to communicate to mobile host. The mobility features must be supported at mobile host and its mobile gateway (Mobile IP terminology Mobile Node and Home Agent).

Requirements on mobility:

- The solution must deliver IP mobility, thus keeping the same IP address while changing the access networks (see 2.8). IP mobility allows the transparent use of all IP based protocols, without any need for specific changes in previously developed applications.
- Mobility must be independent from the link layer. Changing the different types of access networks, such as from UMTS on WiFi, must not interrupt the connection insofar as IP connectivity is available.
- The mobile host is unaware of its PoA, thus its Internet IP and port. The only way of detaching the PoA is executing proactively an update procedure in some intervals. The procedure updates also the current PoA at the participants.
- The bandwidth must be used efficiently. The signalling packets should be proportional to the network changes and host movements. When the host moves rarely the signalling must be low, and vice versa.

Security requirements:

- The mobile signalling and application data must be: authenticated, encrypted, replay and integrity protected.
- The session must not be traceable to IP address. Header information, such as session ID's, can be static during the session. This allows an attacker to map certain sessions to PoA. The IP addresses are physically allocated and the attacker can trace the physical movements of the mobile host in this way. This must be avoided in M-VPN.
- The principle of Perfect Forward Secrecy must be implemented, see A.2.2.6.3. The compromising of a single private key must not lead to the compromising of other keys.
- The solution must deliver security between the Mobile Host and the Gateway. The intermediated devices are not considered trustful. Solutions, like hop-by-hop protection implemented in SIPS [2], do not deliver the required security. In the hop-by-hop solutions, the connection is only protected between the clients and proxies. The data in the proxies is unprotected, thus in clear text.

The requirements in terms of security and mobility are delivered from a practical experience. They do not try artificially to tie the circle of possible solutions, in order to develop research work. Currently, there is a big gap between the theoretically developed protocols and the real implementation requirements as shown in 2.10. The primary target is to

achieve acceptable mobile solutions, which can be implemented in the current Internet structures.

2.8 Principles of mobility

Network mobility means changing significant network parameters and at the same time keeping the communication sessions active. Significant network parameters are usually the device (service) identifiers, such as the MAC address, IP address, TCP or UDP port, URL etc. When some of the identifiers change, the data cannot be delivered to the destination. To overcome this situation, there are two alternatives:

- First alternative is to inform the communication node of the new identifier. This requires certain features at the node, like redirection in HTTP protocol.
- Second possibility is to map the current identifier to the original one at some intermediate node.

The first alternative is not suitable, since the mobile nodes will not be able to communicate with the majority of the Internet hosts, which do not support the required features. The second possibility is implemented in the mobile protocols. The mobility protocols create dynamic mapping on some of the communication elements, like end hosts or intermediate devices. The change is transparent for applications and does not require any special features. Depending on the mapped identifier, the protocols can be classified in different groups. For example, Mobile IP deals with mobility on IP layer.

2.8.1 Layer of delivery and operation for the mobility

The mobility must be defined regarding the layers in first place. *Layer of delivery* denotes the transparent layer enabled by the protocol. The applications benefit from this transparency. For example: constant IP addresses means delivering the network layer, like in Mobile IP. The mobile protocol operates at certain layer, called *Layer of operation*. Most of the mobile protocols operate at application layer, like Mobile IP where the signalling is done using an application layer. It can be achieved using lower layers, like in 3G networks.

In this thesis, the target is to achieve network layer of delivery, called IP mobility. There is simple principle: the lower the layer, the more protocols are served. For example: FTP Clients must not be rewritten if the network layer and above do not change.

Which is the optimal layer of operation for mobility protocol is an important question for the design. It can be answered by discussing the typical layers for home broadband connection, see Figure 2.7. The principle is the same even there can be different layers, like in 3G networks. The operation layer should not be interrupted. Otherwise, the intermediate device must support the protocols. For example, the NAT device interrupts the transport layer, since it manipulates the ports in TCP/UDP. When the protocol operates at transport layer, NAT device must support its header. The device must handle the header in an equivalent way to TCP, UDP, ICMP etc. Without supporting the protocol, the packets will be dropped.

Bringing new features to the intermediate devices requires investment and a lot of non-technical organisation work, which can take several years to set up. This is not a realistic strategy in the heterogeneous Internet. For this reason, the mobility protocol must operate at application layer.

The mobile protocol must operate at application layer and deliver a constant network layer. The controversy exists in that the higher layer must deliver lower layer features. Normally it is exactly the opposite: the features of the lower layer enable higher layer properties. This problem is solved by tunneling described in the following chapter.

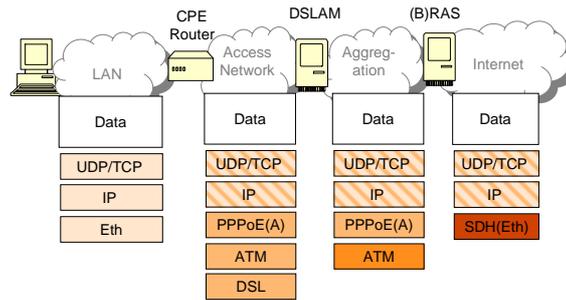


Figure 2.7: Layers typical broadband connection

2.9 Tunneling for enabling mobility

The controversy of the last chapter that the higher layer must deliver lower layer features, can be solved by using tunneling. The basic idea is to encapsulate parts of the packet again in new packet. This can be achieved at every layer and two examples are shown in Figure 2.8. The original packet, in white in the figure, is transported in the application layer of GRE or ESP packet, see a) and b) in Figure 2.8. The receiver of the tunnelled packet must decapsulate and forward the original packet to the application. The connection is transparent for the application. The mobile solutions use always some type of tunnel. The tunnel itself has some disadvantages:

- It brings an overhead of packets because of the encapsulation. Increasing the overhead decreases the useful data, which can be transported in one packet.
- A second disadvantage is the sub optimal routing. The packets are routed to the tunnel destination and then to the inner packet destination. This is not the shortest (optimal) path.
- The tunnel causes some mixture of the layers, so the IP or UDP header can be met twice in the same packet. The packet structure becomes complicated and some inter-layer relations cannot be always treated correctly.

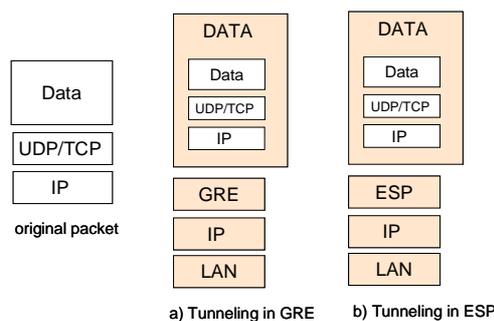


Figure 2.8: Tunneling

Tunneling is unavoidable in the mobile protocols, which operate at the application layer and deliver network layer functionalities. For Example, the Mobile IP [4] operates at application layer and uses tunneling for delivering constant IP.

2.10 Existing approaches for secure IP mobility

Many efforts are carried out focusing on mobility. The protocols claim to solve the mobility problem that is investigated in this section. They deliver partial security according to 2.6 in combination with security protocols. The solution candidates are classified into groups, depending on their properties:

- Combination of mobile protocols and security protocols at different layers
- Security with mobility features in the same layer
- Protocols at research stage with new ideas

2.10.1 Mobility and security at different layers

A common idea is to separate the mobility and security in two different layers. The advantage of this solution is to use already existing protocols without any new developments. The two-layer principle is the same for all mobile protocols- one layer for mobility and one for security. The most popular representative is Mobile IP [4, 6] with IPSec [13]. Other mobility protocols such as Network Mobility [10], Hierarchical Mobile IP [11] and Proxy Mobile IP [52] are briefly overviewed, since they do not change the working principle and the conclusions. Overlaying of mobility protocols with other security protocols such as SSL, SMIME does not change the working principle. These change the security layer and are application specific, thus more restrictive. They are not presented here, since IPSec delivers the most straightforward implementation.

2.10.1.1 Mobile IPv4 overview

Mobile IP (MIP) is a popular protocol for achieving IP mobility. The protocol is specified for IPv6 in [6] and for IPv4 in [4]. The focus in this section is on Mobile IPv4. The Mobile IPv6 is described in 2.10.1.4.

The key elements and the terminology are provided here to align all readers. Details on the protocol can be found in [4, 6]. The hosts (nodes) in Mobile IP are classified as:

- *Mobile Node* (MN) is a host or a router that changes its point of attachment from one network to another. It obtains a new IP address by every change.
- *Home Agent* (HA) is a router on a Mobile Node's home network. It tunnels the datagrams for delivery to the Mobile Node when it is away from home. It maintains the information of the MN's current location.
- *Foreign Agent* (FA) - A router at a Mobile Node's visited network, which provides routing services to the Mobile Node in the visited network.
- *Correspondent Node* (CN) - An Internet node without any mobile functionality, which communicates with the MN.

The Mobile Node (MN) has two IP addresses: care-of address (CoA) and home address. The home address is constant and used to communicate with the other hosts (Correspondent Nodes). This address is part of the Home Agent network. The care-of addresses are the transport IP's assigned by the visited network. This temporary address is used to build a tunnel from Home Agent to Mobile Node. The original packet is encapsulated in the IP packet with care-of address. The general structure is shown in Figure 2.9.

The packets sent to the home address of the Mobile Node are intercepted by the Home Agent (HA), step 1 in Figure 2.9. The HA tunnels the packets (IP in IP [8]) to the current location of the Mobile Node, step 2 in Figure 2.9. The Mobile Node or optional the Foreign Agent decapsulates the packet. The Mobile Node (MN) can reply directly to the

Correspondent Node, step 3 in Figure 2.9. Optionally, the MN can use reverse tunnel to the Home Agent [7]. The HA decapsulates and forwards the packets to the CN. The Foreign Agent can also perform encapsulation of the tunnel instead of the MN. When the MN changes its PoA it notifies the HA of its new care-of-address.

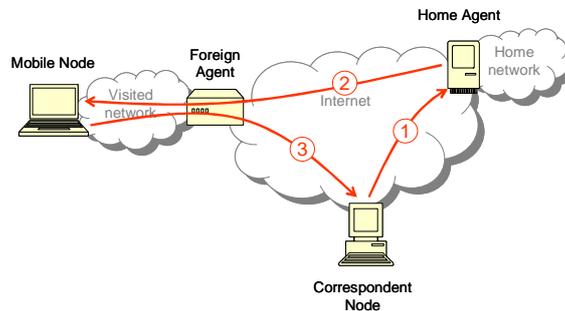


Figure 2.9: Mobile IPv4

The Mobile IP (MIP) provides integrity protection of binding updates during the registration. The signalling in general is not confidential. The confidentiality is a retirement for mobile IP security as described in 2.6. Protection of application data is out of scope of Mobile IPv4, so this must be done by some other protocol, such as for example IPsec. The possibility for protection of signalling and application data with IPsec is presented in following chapters 2.10.1.2 and 2.10.1.3.

The main limitation of Mobile IPv4 is the assumption that the Mobile Node is aware of its PoA and changes of it. The Mobile Node must send binding update by every PoA change. The contrarily was clearly shown in 2.4 for NAPT environments and in 2.5 for multi-homed hosts. Therefore, Mobile IPv4 cannot work in the current Internet structures with NAPT and on multi-homed hosts.

2.10.1.2 Mobile IPv4 over IPsec

The first possibility is to protect Mobile IP (MIP) with IPsec as under layer. The constellation is shown in Figure 2.10. IPsec session is established from the MN to the IPsec Gateway. The IPsec Gateway is in front of the Home Agent from MN perspective. The MN can register and send packets to Home Agent (HA) through the IPsec protection. The IPsec tunnel is point 2 in Figure 2.10, which protects the Mobile IP protocols at point 3. Reverse tunnel [7] must be used in this case, so that the packets are transmitted through the tunnel.

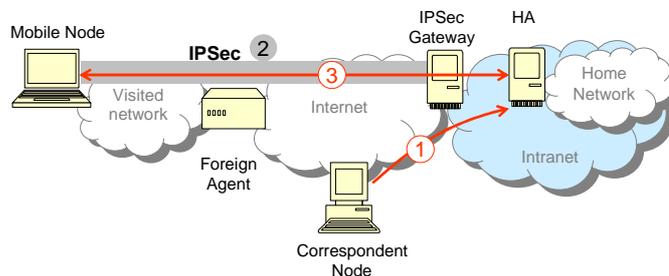


Figure 2.10: Mobile IP over IPsec

The HA is logically behind the IPsec gateway denoted as the Intranet. An Intranet could even have private IP addresses, which are not routable from the Internet. The mobile signalling is encrypted, integrity protected, and authenticated which was a necessary point for secure IP mobility as defined in 2.2. The Mobile IPv4 does not implement confidentiality (encryption) of the signalling.

If the IPSec Gateway and the HA are implemented in different devices, then IPSec in Tunnel mode must be used. The Tunnel at MN must use the care-of address as a source IP, since this is the only routable address. If the IPSec Gateway and the HA are implemented in the same device (same IP) then transport mode can be used. In transport mode, there is not need for intranet IP addresses.

There are two major problems using Mobile IP over IPSec regardless of tunnel or transport mode of IPSec operation:

- IPSec uses the care-of address to establish the IPSec session. This address is temporary and changes every time the mobile host moves to a different foreign network. This means that when changing a visited network, a new IPSec session must be established. This is practically a controversy to mobility – the IPSec session is static. The reestablishment requires resources and time. If host is moving very frequently, it may even occur that there is not enough time to negotiate the IPSec session.
- The Foreign Agent cannot be used in the registration, since the traffic between the MN and IPSec gateway is encrypted. The FA cannot inspect the headers as defined in [4]. The FA must not be involved. Use of hop-by-hop protection, thus one IPSec session from MN to FA and second from FA to HA, is considered as unsecure as defined in 2.7.

To recapitulate, the Mobile IP over IPSec is not a reasonable alternative, since the IPSec session must be re-established with every movement of the mobile host. Furthermore, no Foreign Agent can be used.

2.10.1.3 IPSec over Mobile IPv4

The security protocol can be set on the top of IP mobility. The representative case is shown in Figure 2.11, where the IPSec gateway is behind the HA. The Mobile IP (MIP) session is established, see point 2. The IPSec is transported in the Mobile IP tunnel, see point 3. The HA forwards the IPSec packets to the IPSec gateway.

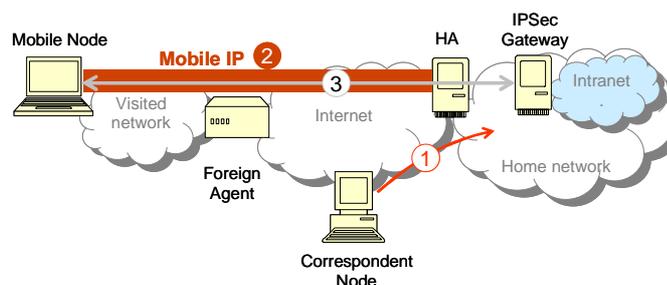


Figure 2.11: IPSec over Mobile IP

The Intranet, which is accessed via IPSec, is part of the home network. This is exactly the inverse case to the one described in 2.10.1.2. From a routing perspective, this case can be split into three sub cases, which are:

- 1) The HA and the IPSec gateway can be installed at the same host. The HA and IPSec gateway will be the same node with the same IP address. The IPSec must not be encapsulated in Mobile IP tunnel and a lot of overheads will be saved. The IP-in-IP tunnel [8] described Mobile IP [4] can be replaced by IPSec tunnel.
- 2) HA and the IPSec are different hosts. This is presented in the Figure 2.11 and taken as an example. The IPSec must be encapsulated in the Mobile IP-in-IP tunnel. This multiple encapsulation leads to a lot of overheads (Figure 2.12) and it is not very comfortable. The outer IP header (CoA in Figure 2.12) contains the care-of address

of MN and the HA's address. The packet can be routed between the MN to HA, see Figure 2.12. The second IP header (Ha) contains the home address of the MN and the IPsec gateway address. In this way, the IPsec gateway communicates with a constant Mobile Node home address. When IPsec is used to access to corporate network, then the IPsec can be in tunnel mode. The inner IP header (Ca) contains the corporate IP addresses. There are three IP headers and two UDP/TCP headers in this encapsulation. This is obviously not ideal and significantly reduces the real data payload. In practical terms, there are two tunnels IP-in-IP and IPsec tunnel. The IPsec session must not be re-established when the MN attaches to a new access network.

- 3) The IPsec gateway can be on the Internet. Consequently, the Intranet is not part of the home network. The HA decapsulates the tunnelled packet from the MN and forwards the IPsec packet to the IPsec gateway on the Internet. The whole IP packet must be encapsulated in the same way as shown in Figure 2.12. The return transport is exactly the opposite.

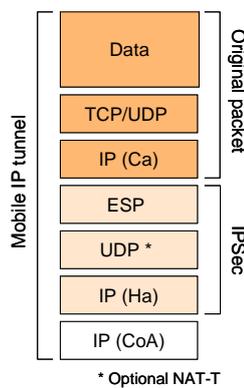


Figure 2.12: IPsec in MIP

IPsec secures the data exchange between the Mobile Node and the Home Agent. The IPsec session is carried out in tunnel IP-in-IP or MIP. The advantage of this type of implementation is that the IPsec must not be re-established when the user changes its IP.

The main issue is that the Mobile IP itself is not protected by IPsec. The IPsec is over Mobile IP and cannot protect the lower layer. A man in the middle attacker can read passively the mobile signalling. The registrations are authenticated in Mobile IP and cannot be manipulated but in clear text. A major requirement for secure IP mobility is not fulfilled and therefore, the solution is not acceptable.

2.10.1.4 Mobile IPv6 and IPsec protection

The Mobile IPv6 [6] is designed for working in IPv6 [5] environments, as the name says. The Mobile IPv6 profits from the gained experience of IPv4 networks. It takes also advantage of the new IPv6 features, such as the additional header extension for optimal routing, called routing header option. Details for Mobile IP can be found in [6]. The node types are shown in Figure 2.13. The major differences to Mobile IPv4 can be found in RFC 3775 [6] and can be summarised as:

- There is no need for a Foreign Agent (FA). Mobile IPv6 operates without any special support at the visited network which is a big advantage for the deployment.
- Mobile IP is part of the standard IPv6 stack and mobility functionality can be expected in every native IPv6 host. Most of the packets are sent direct from the Mobile Node to the Correspondent Node using direct mobile binding.

- Route optimisation is part of Mobile IPv6 and standard feature. The route optimisation in IPv4 is usually blocked by the anti-spoofing (ingress filtering) at the ISPs. The Mobile IPv6 allows coexistence of route optimisation and anti-spoofing filters.
- To increase the security of the direct binding, the return routability procedure is defined.
- Mobile IPv6 is independent of the link layer, because it uses the Neighbour Discovery [18] of IPv6 instead of ARP mechanism of Mobile IPv4
- The IPv6 encapsulation (and the routing header) removes the need to manage tunnel state in Mobile IPv6.

The mobile host receives its global care-of address through the IPv6 address configuration. The IP configuration can be stateless [19] or statefull [20]. Then, the Mobile Node builds an association with its Home Agent, called binding. The binding is a mapping of MN's care-of address to the home address for a certain time (lifetime).

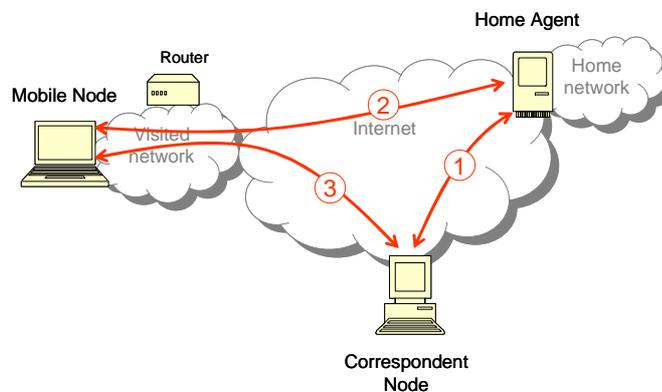


Figure 2.13: Mobile IPv6

The Home Agent adds an extension routing header type 2 to the IP packet [6], when sending a packet to the Mobile Node. The packet structure is shown in Figure 2.14 a), where MIP denotes the mobility header. The field msg. data in Figure 2.14 is the mobility message sent in MIP. The routing header type 2 (route. ext in Figure 2.14 a) specifies the home address of the Mobile Node. The destination option (dest.opt.) of the IP header (home address opt) is used by the sending packet from the Mobile Node to the Home Agent. The option contains the home IP address of the Mobile Node. The dest.opt. is not used when the HA sends to the MN. The IP header has the source IP of the Home Agent and the destination of the Mobile Node care-of address (CoA).

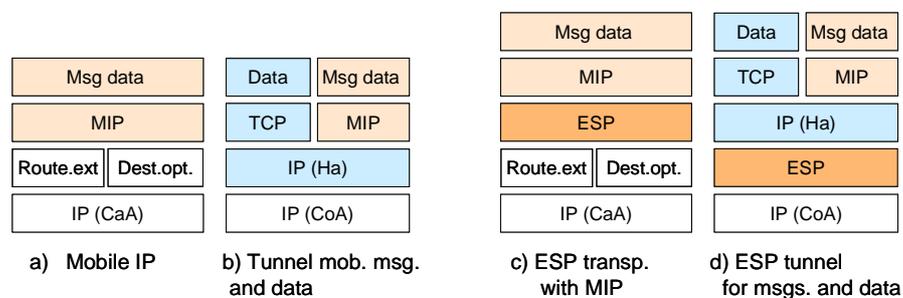


Figure 2.14: Mobile IP header structure with and without IPSec protection

The Home Agent intercepts all packets sent from the Correspondent Node to the Mobile Node and tunnels them in IP-in-IP [21] to the care-of address, when the Mobile Node is away

from its home network (step 2 in Figure 2.13). The mechanism is quite similar to the Mobile IPv4 tunneling. The packet structure is shown in Figure 2.14 b).

The Mobile IPv6 is part of the standard protocols stack and mobility support can therefore be expected by all Correspondent Nodes. The main resultant advantage is the routing optimisation. The Mobile Node and the Correspondent Node can establish binding and exchange packets directly with each other, step 3 in Figure 2.13. The binding is established using a return routability procedure. In general, the procedure conducts a simple check against spoofing. The binding messages are transmitted in two redundant ways: directly from the Mobile Node to the Correspondent Node and the secondly and indirectly through the Home Agent. The binding is established only when both messages are received. The binding messages are tunnelled through the Home Agent as shown in Figure 2.14 b).

There are possible two IPSec sessions in Mobile IPv6: (1) Application data protection between the Mobile Node and the Home Agent in the case of Tunneling, see section 15.7 of [6]. (2) Protection of the Mobile IP signalling from Mobile Node to the Home Agent described in [22].

The ESP in tunnel mode replaces the IP-in-IP tunnel by the protection of the data exchange (step 2 in Figure 2.13) between the Mobile Node and the Correspondent Node through the Home Agent. It is also used by the return routability procedure. The payload structure is presented in the Figure 2.14 d). The ESP header is over the IP header and protects the Mobile IP messages and the application data from/to the Correspondent Node.

ESP protects the mobile signalling messages between the Mobile Node and its Home Agent in transport mode shown in Figure 2.14 c). The binding updates are in the payload of ESP protocols. There is no need for tunnel mode, since the receiver of the IPSec packet is also the receiver of the end Mobile IP packet.

The exchanged data and the mobility signalling are protected according to the Mobile IPv6. The prime target for authentication, encryption, and integrity is met in this way. Unfortunately, the current specification of ESP [14] cannot cover this type of implementation as described in [22]. The source and destination IP addresses in one ESP communication must be constant as described in A.2.2. Every change to the IP addresses or port during the session causes disconnection. The IPSec does not include any mechanism for address update and relies on the constant IP parameters. In practical terms, a new ESP session must be established as by Mobile IPv4 with every change of the care-of address.

The authors of Mobile IPv6 suggest in [22] changes to the IKEv1 [15] and ESP [24] implementation to overcome this. The basic idea is that Mobile IP must control the IPSec SA and SDP parameters and change them depending on the Mobile Node movements. With every change of the care-of address, the Home Agent updates the new address in the current IPSec SA and SPD. The changes can be summarised as:

- When the Mobile Node receives ESP packet, the route header type 2 must be considered as the destination of the packet. In this way, the received packet can be preceded by the selectors of the SPD.
- The SPD policy must be deactivated when the Mobile Node returns to his home network, thus no IPSec.
- When the Home Agent sends packets to the Mobile Node, the correct care-of address must always be set by IPSec. IPSec cannot handle changing care-of address. There are two possible solutions:
 - IPSec stack works with the home IPs, which is constant. The packets are intercepted after the IPSec encryption and the care-of-address is set.

- Direct API interaction with the SPD and SAD, thus IPSec becomes mobile.
- When the Mobile Node sends IPSec protected binding messages to a Correspondent Node, the IPSec must use the right addresses. It is the same problem as the previous point.
- When the Mobile Node receives a changed set of prefixes from the Home Agent, a new security policy must be configured.
- The Home Agent must consider the routing option of the received packet as the source address of the packet in order to match the SPD.
- The ESP header does not protect the outer IP header with the source and destination IP. Possible attacker can manipulate the outer header. The binding IP must be included in the ESP payload and thus protect it. This is done with the use of a destination option in the binding updates and mobile prefix solicitation.
- The Home Agent must take significant care of matching the IPSec ID and the mobile IP bindings. Mobile node must be prevented by using its security association to send a Binding Update for another node. The Home Agent must carefully match the IDs in the two layers.
- Circular dependencies must be avoided. For example: An IKE session is required for proceeding a binding update and in the same time to establish IKE session is required a binding update.
- The Home Agent and the IPSec Gateway must be implemented on the same host, see 2.10.1.3

The Mobile IP and IPSec are tied in one implementation - they cannot be implemented in two independent layers. The IPSec stack must be changed in order to work with Mobile IP, because Mobile IP should control the IP addresses in the SAD and SPD. Changing the IPSec implementation stack has influence on all other protocols using IPSec. Even there are not currently implementations of IPSec with Mobile IPv6 the following problems can be pointed out:

- The changes of IKEv1 to support Mobile IP lead to interoperability problems. There will be two implementation types: with Mobile IP and standard IKE. This is considered as potential source of many failures.
- The movements of the Mobile Node can be easily traced. The IPSec session uses a constant SPI, which can be matched to physical care-of-addresses. The target of the secure mobility is also to hide the movements of the host between the access networks.
- Matching the IDs between the IPSec and Mobile IP is very important. The SPD entries must define which credential of IKE phase 1 can be used to protect certain Binding Updates. Otherwise, session hijacking attacks can be carried out by insider. The Mobile IP must authorise the IKE ID for certain mobile binding. The only possible way to implement this is the inspection of Mobile IP packets by the SPD selector. After the packet is decrypted, the SPD verifies that the packet matches the policy defined by the selectors (see A.2.2.4). At this stage, an inspection of the Mobile IP binding can be made. This requires joined database between IPSec and Mobile IP, which can be very difficult.

- Every time the Home Agent changes its prefix, a new security association must be negotiated.
- When a host is used by multiple mobile users, they can use the same IKE phase 1, since IPSec protects the communication between the hosts. The authentication credential used in IKE phase 1 will be the same for all user of this host. The same IKE credentials will be used to send bindings for different users. The Home Agent cannot determine a misuse within this group of users and this is a theoretical issue. To overcome this issue, for every user different ID in IKE phase 1 must be used.

The IKEv2 [23] is the successor to IKEv1 and is much closer to mobile IP requirements than IKEv1. The IKEv2 and Mobile IP must be tied together even using new version.

The Mobile IPv6 with IPSec is certainly much closer to the required secure mobility than the Mobile IPv4 solution. Unfortunately, IPv6 is not currently implemented on a large-scale basis by the ISP and the mobile operators (Status Q1 2009). The operators are currently evaluation the all-IP backbones. The practical experience on global IPv6 backbone is far less than IPv4. There is still no reliable road map by the ISPs for IPv6 implementation.

2.10.1.5 Hierarchical Mobile IPv6 with IPSec protection

The Mobile IPv6 have been designed for providing mobility at heterogeneous Internet, like using smart phones with constant IP addresses instead of phone numbers in CDMA2000 [26]. Implementing Mobile IP in large backbones and especially in wireless networks arise the following issues [11]:

- Every time the Mobile Node changes its care-of address, it must send binding updates to the Home Agent and optionally to all Correspondent Nodes, when not using tunneling. The binding update requires 6 packets [6] according the return routability procedure. This reflects in delay of about 2 times the round-trip (if all nodes have the same time distance between them and the Home Agent registration is done in parallel to the Correspondent Node updates). This handoff delay decreases the performance of the Mobile Node connection and running application. Supplementary, it generates loads on the access network, which is the most expensive part of the network transport.
- The Mobile IP structure is flat and strongly concentrated at the Home Agent. The Mobile Nodes always update their binding to the Home Agent, which is the central instance. To avoid load concentration at one node, there is a need for logical distribution between the nodes part of the same administrative domain.

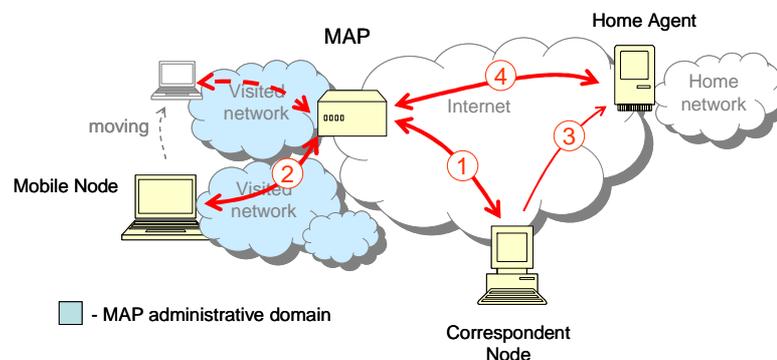


Figure 2.15: Hierarchical Mobile IP

Hierarchical Mobile IP was defined in order to solve these issues and improve the Mobile IP performance [11]. The Hierarchical Mobile IP is add-on to Mobile IP, which adds new features to the current Mobile IPv6 specification.

A new node type is introduced, called Mobility Anchor Point (MAP). The new node functions as a local Home Agent. It keeps a track of the current location of the Mobile Node and tunnels the packets from the Correspondent Node to the Mobile Node. It can be compared to the Foreign Agent in Mobile IPv4. The main difference is that it is an optional element and must not be presented at every visited network. The Mobile Node is free to use it or not. The Figure 2.15 presents the structure. The Mobile Anchor Point (MAP) is a back-to-back user agent. From a Mobile Node's perspective, it serves as a Home Agent. The MAP is the Mobile Node from the perspective of the Home Agent and the Correspondent Nodes. It can be compared to Proxy, thus back-to-back agent.

The Mobile Node (MN) registers at the MAP, shown at step 2, Figure 2.15. The MAP cannot authenticate the MN. On behalf of the Mobile Node, the MAP sends a binding update to the Home Agent (step 4, Figure 2.15). Then it tunnels all packets between the Correspondent Nodes and the Mobile Node (step 2, Figure 2.15). The Home Agent makes a tunnel to the MAP node and MAP to the Mobile Node. When the Mobile Node is moving in the administrative domain of the MAP, the binding is not forwarded to the Home Agent, since from HA perspective there is no movement at all.

The MAP is an intermediate device breaking the flat Mobile IP structure into a hierarchical one. This significantly reduces the number of binding updates at the access network. Furthermore, it hides the exact location of the Mobile Node from the Correspondent Nodes. There is additional delay, because of the two steps forwarding "Correspondent Node -> MAP -> Mobile Node".

To secure Hierarchical Mobile IP with IPsec is a difficult task. As already mentioned, the MAP is Home Agent from MN perspective. The MN establishes IPsec connection to the MAP node. The MAP node can itself establish IPsec to the Home Agent. The MN must trust the MAP node. MAP is part of the access network and probably of the control of not trusted authority, like HotSpot in a hotel. If the MAP is part not trusted, the mobile mode must not use the benefits of the MAP node. In this case, it will function like a standard Mobile IP v6. The MAP and MN must have trusted relation in order to protect the communication with IPsec. In this case, the properties described in 2.10.1.4 are valid.

The MAP should be in the same trusted domain as the Home Agent. In this case, the security issues are the same as Mobile IPv6 with IPsec. If MAP is not part of the trusted domain, then there is no sense in building IPsec sessions to it. On the other hand, the whole point of MAP is not to be in the same trusted domain. Hierarchical Mobile IP makes no significant difference to the security mechanism to the one in Mobile IPv6.

2.10.1.6 Proxy Mobile IP

Proxy Mobile IP enables mobility to hosts without any mobile features, like Mobile IP. The mobility is delivered by the network and not by the host itself, thus the intelligence is at the network and not at the end device. The protocol is specified in [52] for IPv6 and in [53] for IPv4. The motivation for the protocol is mainly 3G networks (cellular networks) where few operators serve large physical areas. The protocol will allow the operators to assign constant IP to the cell phones, which can be used for VoIP in NGN. The advantage is that the operators can guarantee service delivery without changing the end devices.

The main properties are described in the following paragraph and then analysed regarding Mobile VPN usage. More details can be found in [52, 53]. The Proxy Mobile IP protocol bases on Mobile IP and defines two new key elements:

- *Local Mobility Anchor (LMA)* is the Home Agent for the Mobile Node in a Proxy Mobile IP domain. It is the topological anchor point for the Mobile Node's home network prefix(es) and is the entity that manages the Mobile Node's binding state.
- *Mobile Access Gateway (MAG)* is a function on an access router that manages the mobility-related signaling for a Mobile Node that is attached to its access link. It is responsible for tracking the Mobile Node's movements to and from the access link and for signaling the Mobile Node's local mobility anchor.

The Mobile Node (MN) establishes a point-to-point connection the access router with mobility functionality (MAG), step 1 in Figure 2.16. The MAG authorizes the MN node if it can be served with mobility and which is its Home Agent (HA). At next step 2, the MAG connects the Local Mobility Anchor and performs Proxy Binding Update. The LMA has local HA functionality and manages the MN bindings. The LMA may connect the HA for obtaining the Home Address of the MN at step 3, if it not locally cached. On successful Proxy Binding Update, the MAG assigns to the MN its Home Address. The Mobile Node uses its home address for communication, thus the MN does not have any Care-of-Address. For communication to the MN, the Correspondent Node sends the packets to the HA. They are forwarded to LMA, MAG and the MN. If the Correspondent Node has Mobile functionality, it may communicate directly to LMA as described in Mobile IPv6.

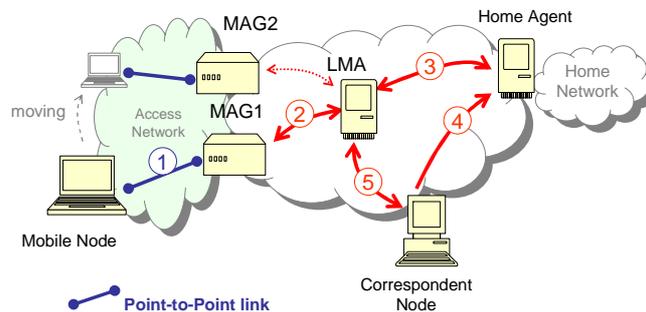


Figure 2.16: Proxy Mobile IP

The main idea is that to assign the MN its Home Address and in this way, there is no need of any mobility functionality at end host (no Care-of-Address at host). This requires that the access router must support mobility functionality. This can be achieved when single operator controls the network, like a 3G operator can enroll mobile functionality to all access routers and control them.

Proxy Mobile IP requires functionality of the access network, thus intelligent network. A main requirement on Mobile VPN, see 2.6, is that the mobility must not require special features of the intermediate network devices, like routers and switches. The Proxy Mobile IP has the scope of 3G backbone solution and not general Internet solution. There is not realistic to consider that the subscribers of WLAN Hotspot and DSL are going to enable the feature. The Proxy Mobile IP is not considered further for Mobile VPN, since it does not targets heterogeneous Internet.

2.10.1.7 Network Mobility (NEMO) with IPSec

The network mobility protocol (NEMO) [10] treats networks and not single hosts, as the Mobile IP does. It can practically manifest itself in the example of a moving train. The hosts inside the train can be the laptops of the passengers. The hosts in the train are static to each other and part of one network. The network is moving through different access networks.

The protocol is extension of Mobile IPv6 and its global structure is shown in Figure 2.17. The hosts in the mobile network do not need any mobile functionality and are not aware that they are part of the mobile network.

The Mobile Router (MR) is a new node type, which registers at HA in a similar way as the Mobile Node does. The MR registers not a single address, but one or multiple network prefixes (Step 1, in Figure 2.17). A Correspondent Node can send packets to a host part of the network (Step 2, in Figure 2.17). The packets are intercepted by the Home Agent and forwarded in tunnel to the mobile network (Step 1, in Figure 2.17). There is no direct binding between the Correspondent Node and the mobile router, thus all packets are routed through the Home Agent. The protocol makes minimal extensions to the classical Mobile IPv6.

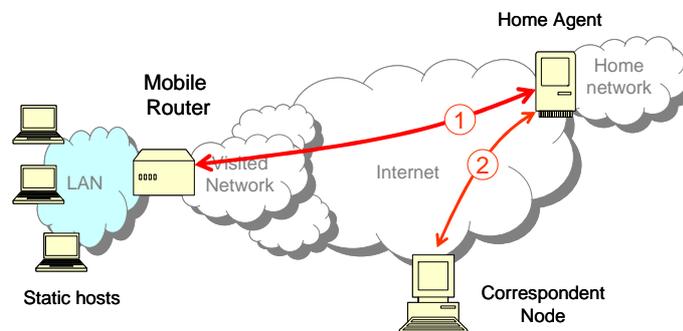


Figure 2.17: Network mobility

In fact, the same result as by Network Mobility (NEMO) can be achieved with the classical Mobile IP. The mobile host must simply run routing (static or dynamic routing protocol) and the Home Agent must treat the tunnel as an interface.

The tunnel between the Mobile Router and the Home Agent is built with Mobile IPv6. The security mechanism is the same as in Mobile IPv6 described in chapter 2.10.1.4.

2.10.1.8 Conclusion of two layer solutions for secure IP mobility

The IPSec under Mobile IPv4 [4] cannot be used to protect the signalling and the application data, since a new IPSec session must be negotiated with every change of the care-of address. When using IPSec over Mobile IP the mobile signalling is unprotected and can be manipulated.

Concerning Mobile IPv6 [6] can be concluded that the IPSec policy must be changed. This requires changes in the IPSec stack, which can cause interoperability problems. The main problem with Mobile IPv6 is that it requires IPv6. There is no currently reliable time schedule for IPv6 deployment with the major Internet service providers.

A general problem for IPSec is that the connection can be traced. The SPI number in the ESP header is constant during the session, so it can be mapped to a care-of address. The physical movements of an unknown user can be traced in this way.

The two layers solution cannot meet the requirements of chapter 2.6. The problem is of a principle nature. The security protocol is defined for static IP addresses, thus it cannot be located on the layer under the mobility. If security is over the mobility layer, it cannot protect

the mobility signalling. This dilemma cannot be solved without bringing mobility at the security layer and vice versa.

2.10.2 Secure protocol with mobile features

A different approach is to integrate the mobility and security in one protocol, thus single layer solution. The development of IKEv2 [23, 27], ESP [24] and AH [25] opened new possibilities for secure IP mobility. The main reason for the new standardisation was the big gap between the real implementations and the standard of IKEv1 [15] and ESP [13] as described in A.2.2.10.

2.10.2.1 Overview of IP Security v2 and Internet Key Exchange v2

IKEv2 [23] and ESPv2 [24] do not specify new features in the first line. Most of the features were already existing RFCs or draft standards. The main differences relevant for mobile security are: new database type, simplified negotiation, more authentication algorithms, new processing rules, NAPT support, compression algorithm, and encryption methods. Without going into deeper details, they are shortly explained in the following paragraphs.

A new database, called Peer Authorization Database (PAD) [27], was created. It enables the interaction between the SPD and the IKE. The database:

- Identifies the peers that are authorised to use IPsec entity
- Gives the possibility for specification of authentication protocol for each peer
- Provides the authentication data, for example: passwords
- Defines the IPsec SA, which can be created based on the IKE session
- Optionally contains information about the location, such as the IP address or DNS entry

The PAD is a reflection of the current complexity of the authentication policies. One IPsec gateway serves multiple administrative domains such as the user, guest, special departments etc. All of these groups have different authentication algorithms and access rights. Together with the SPD, the PAD database builds something like a higher layer firewall with different policies.

The second new extension, which is relevant for the secure mobility, is the method of matching the inbound ESP packets. By processing inbound traffic, SA matching can be made with [27]:

1. Combination of SPI values, destination address and source address
2. Both SPI and destination address
3. Match on SPI only

The closer match has the highest priority. This brings advantages for the mobility, because the source/destination IP address is not decisive.

The IKEv2 reduces the number messages and the protocol states. This improves the mobility characteristic of the protocols through increasing the probability for successful establishment. The communication consists of request and response messages for every exchange. The IKEv2 does not differentiate in phase 1 and phase 2. The IKEv1 was designed to work with different protocols, but this idea has failed. The IKEv2 does not try primarily to support protocols other than IPsec (ESP, AH). Consequently, there are no phase 1 or phase 2.

The negotiation has different exchanges. The `IKE_SA_INIT` exchange negotiates cryptographic algorithms, exchanges nonces, and does Diffie-Hellman Key Exchange (DH). The following exchange, `IKE_AUTH`, authenticated the previous messages, exchange identities and certificates. The `IKE_AUTH` messages are encrypted and their integrity

protected with keys generated in the IKE_SA_INIT exchange. The identities are hidden from intermediate attackers. The IKE_AUTH creates also the first IPsec SAs (inbound and outbound). It corresponds to the phase 2 (Quick Mode) in the IKEv1. The CREATE_CHILD_SA exchange generates supplementary IPsec SA (in IKEv2 phase 2). The exchange starts after finishing the IKE_SA_INIT and can be in parallel to IKE_AUTH. It generates new key material with Diffie-Hellman. It sets the protection algorithm and defines the connection selectors. The selectors correspond to the ID payloads in Quick Mode in IKEv1.

IKEv2 supports multiple authentication possibilities. For example, the initiator can use password based EAP [31], like EAP-MSCHAPv2, and the responder can use the digital signature with certificate [32, 33]. This is a great advantage, since the initiators (client) do not usually have a certificate.

The following Figure 2.18 shows packets exchange, where the initiator authenticates via EAP-MSCHAPv2 and the responder with digital signature. In the first packet, the initiator sends Security Association proposals (SAi1) for the protection of IKE negotiation, see Figure 2.18, packet 1. The responder replies with the chosen algorithms (SAr1), its DH values, nonce and certificate request. The further communication is protected with the DH key negotiated in the first two messages. In the third message, the initiator sends its ID, certificate request, SAi2 parameter for the IPsec session and the traffic selectors. The traffic selectors are very important, when the IPsec is used in tunnel mode. They contain the information which traffic must be routed in the IPsec tunnel. Notice that the client does not answer to the certificate request (CERTREQ) message, which indicates desire to use EAP authentication (indirect announce for EAP). The responder sends its certificate, signed hash value, EAP value in message 4. The 5th and 6th message transport the EAP authentication. The 7th message is the signed hash of the payloads. The last message is the IPsec SA parameter and the traffic selectors from the server. In general, the EAP payloads are directly transported as described in [31]. More details and different negotiations can be found in [23].

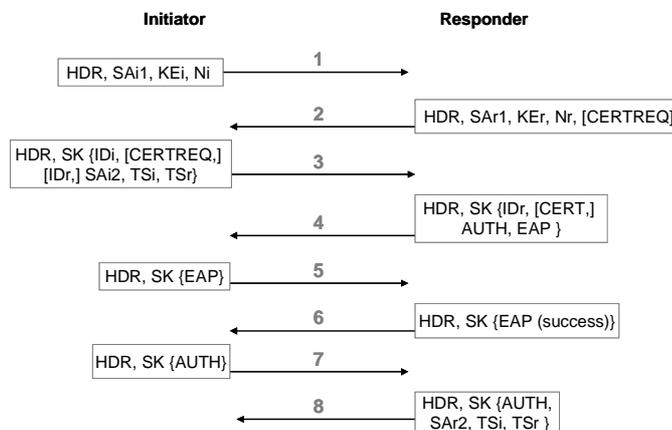


Figure 2.18: IKEv2 negotiation with EAP and digital signatures

The abbreviations for the header are:

AUTH	Authentication
CERT	Certificate
CERTREQ	Certificate Request
SK{ }	Encryption and integrity protection
EAP	Extensible Authentication
HDR	IKE Header
IDi	Identification - Initiator
Dr	Identification - Responder
KE	Key Exchange
Ni, Nr	Nonce

SAi/rSecurity Association
TSi Traffic Selector - Initiator
TSr Traffic Selector – Responder

The next new feature in IKE v2 concerns fragmented packets in the negotiation. The experience with IKEv1 shows that sometimes the UDP packets need to be fragmented. Typically, when certificates are used in the authentication. The certificates with 2048 bit key (currently standard) are commonly bigger than 1500 Byte and cannot usually be transported in a single UDP datagram. The problems arise since the NAT devices can drop the fragments in order to have short transition delays (see A.2.1.9). Furthermore, firewalls treat out of sequence fragments as an attack and consequently drop the fragments too. The IKE negotiation cannot be finished when the packets are dropped. To solve this problem in IKEv2, the peers can send a URL link to their certificate instead of sending the certificate itself. The packets stay small and do not need to be fragmented. The negative effect of this solution is that accessing this link can cost time. When multiple retries are needed, this can cause a timeout by the sender and it will start a new negotiation. A circular dependency without solution will be the result. It is not clear how the application can solve this issue.

The IKEv2 extends the use of tunnel mode with ability for configuration of the initiator and responder. The new feature is motivated by the high deployment of the non-standardised CONF_MOD in IKEv1 (see A.2.2.10.8). The new feature defines "CFG_REQUEST/CFG_REPLY" and "CFG_SET/CFG_ACK" exchanges. The first exchange allows an IKE endpoint to request information from the peer. The peer can assign parameters such as IP address, DNS, WINS etc. This is very useful when using IPsec for remote access, where the client (initiator) does not have a pre-configured IP address. The gateway maintains IP pools, DNS server parameter etc, which are assigned to the client. The "CFG_SET/CFG_ACK" allows an IKE endpoint to push parameters to its peers. The peer can accept this configuration and use it for the tunnel. This parameter can be a static IP address for example. The set configured must be carefully considered, since attacks can be run from inside. The insider attack can try to acquire a higher privilege level by pushing a malicious configuration. Both configuration exchanges are executed before the IPsec SA creation and are usually part of the IKE_AUTH exchange.

2.10.2.2 IKEv2 Mobility and Multi-homing Protocol (MOBIKE)

IKEv2 itself does not provide any mobility support. The MOBIKE [28, 29] protocol is bringing mobility to the IKEv2 [23] and ESPv2 [24]. The protocol defines extensions to the existing IKEv2 specification, which are briefly described in this section.

The MOBIKE defines a new notification payload called UPDATE_SA_ADDRESSES. The extension payload is sent by the initiator to change the IP address in the IPsec SA and IKE SA. The main principle in MOBIKE is that only the initiator of the connection can change IP parameter of the IPsec SA at the responder. After the responder receives this notification, it will change the address of the IPsec SAs. The notification is encrypted with the IKE SA session. The notification cannot be replayed by attacker and in this way, the IPsec session cannot be directly hijacked by attacker. The main problem is that the IP/port is part of the header, which is unprotected see 2.4 and 2.5. The IP/port parameters are extracted from the unprotected packet header. A man-in-middle attacker changes the IPsec SA parameter indirectly. An attacker must intercept and drop some of the IKE SA packets. This will motivate the initiator to send notification. The attacker can manipulate the header of the notification message.

Another MOBIKE extension is the list of alternative peer's IP addresses, called ADDITIONAL_IP4_ADDRESS and/or ADDITIONAL_IP6_ADDRESS. If one of the addresses is not responding, then the initiator can try the next. The IP addresses are

exchanged in the authentication (IKE_AUTH) or with separate notification. This improves the possibility of reaching the peer. For example: if the peer is multi-homed it can certainly send all its addresses. If one of the IP becomes unreachable, the connection can be moved to the new one.

IKEv2 has a peer's status detection mechanism, formally known as Dead Peer Detection (DPD). This is an empty IKEv2 INFORMATIONAL message. It allows to detect if the currently parity is reachable. The Dead peer detection is notation of the IKEv1 [16], in IKEv2 status detection is used instead. The status check is typically done when no data is received from the peer for a long period. The payloads NAT_DETECTION_SOURCE_IP and NAT_DETECTION_DESTINATION_IP are used in the INFORMATIONAL message to detect if the NAT mapping (binding) has changed. The responder of message must include the received packet in the reply, so the sender can compare with the sent one. When the mapping has been changed, the peer must immediately send an update of SA (UPDATE_SA_ADDRESSES). The update can also be send by suspicions that the mapping has changed.

If the peers have multiple addresses, the status check procedure (DPD) can test the paths. Path denotes a combination of the initiator and responder address and port. There are multiple paths when a peer has more than one address. An active path is one in current use. A working path is the path, which can be used for communication. The initiator must be aware of all working paths (address combinations). This reduces the time for switching between the paths, when one of the addresses becomes unreachable. For this purpose, a new procedure "Path testing" is defined in MOBIKE. The initiator tests whether the responder is reachable through certain path. IKEv2 INFORMATIONAL request/response messages are used, thus the same by status detection. Implementations may execute path testing to find alternative or better paths, even there is currently an active path. The procedure for testing the path is called return routability check. The procedure is one of the new extensions and recommended behaviour. It should be executed before updating the IPsec SAs. It may also be executed during the session for any other purpose. The cookie values protect against replay attacks.

These are the major new components of MOBIKE [28, 29] in IKEv2.

2.10.2.3 MOBIKE for secure mobile connections

MOBIKE is making a big step towards improving the security of mobile connection. The IETF working group has understood that the two layers model suggested in Mobile IP is not working. Although its advantages, there are numerous still open issues. The reason for them is the static origin on IKEv2. The mobility extensions are pushed down to the static IKEv2. The core IKEv2 protocol is kept and the extensions cannot overcome the restrictions.

After analysing the specification of IKEv2 and MOBIKE, the author points the following short comings:

- The physical locations of the Mobile Nodes are traceable as described in 2.6. The IPsec SPI parameters are constant during the session and are part of the ESP header.
- The MOBIKE requires about 8 exchanges to establish IPsec SA, see Figure 2.18. This is not optimal for mobility protocol and leads to problems with fast moving devices. The connection can be dropped before final establishment.
- When using certificates for authentication, the connection negotiation can increase by many minutes. As discussed in 2.10.2.2, instead of certificate, the peer can exchange URLs for accessing the certificates. The URLs (web links) use typically HTTP, which has different network requirement (timeout) then mobile protocols. If there are packet losses, the initial download of a

certificate can take minutes. This is unacceptable for the mobile negotiation, where the host can move in this time. The IKE session is dropped before authenticating. There is need for caching to overcome this, which is not part of the implementations.

- The protocol is not optimising the status check messages and nat-keep-alive. The both updates are executed at constant interval (see section 4 in [54] and section 2.23 in [23]) regardless of the network properties and the host activates. The result is wasted resources in not needed updated and long disconnection times. This was main target of Mobile VPN, see 1.3.2.
- There are status checks for every: active IPSec SA, IKE Session and potential path, see [28]. All these checks generate a big amount of updates messages executed at constant intervals in parallel, see 1.3.2. The updates have an overlapping functionality.
- The MOBIKE requires some changes in the IKEv2 protocol in the status synchronisation procedure (formally Dead-Peer-Detection). The change reduces the IKEv2 ability to react properly with NAPT mapping change [28, section 3.8]. When the NAPT mapping is reset, the receiver must not use any authenticated packet to adjust the SA source port and IP parameters. This change in the standard IKEv2 leads to packet loss, when the NAPT mapping is deleted during an active IPSec session. There will probably be two IKEv2 versions: for MOBIKE and static (with and without this change).
- The MOBIKE uses the major principle that only the initiator can change the IPSec SA IP parameter see 2.10.2.2. The result is that the IKE SA parameters can be changed by the both peers and the IPSec SA only by the initiator. This significantly complicates the state synchronisation and can lead to packet loss. Practically, the IP addresses and port in IKE and IPSec SA can be different. Usually the IKE SA uses the address of the last received packet for the response. If the NAPT device deletes the mapping, than the IPSec SA address becomes different to the IKE SA (assuming IKE packet were received from the new binding). All data packet are lost, since the IPSec SA parameters are not correct (showing to the old binding). Only after receiving information notice can the IPSec SA be updated.
- The protocol has a problem when moving between networks with NAPT and without NAPT. Assuming that the communication starts in NAPT-free network, then no NAT-keep-alive is activated. If the host moves to NAPT network during the session then the NAPT mechanism must be activated. The IKEv2 detects the NAPT existence only at the beginning, so in this case the protocol will fail.
- The rule, that only the initiator can change the IPSec SA parameter is unsuitable and it is bad defined. The IP/UDP parameters are manipulated by NAPT devices and (or) by intermediate attack. The IP/UDP values are not trusted by default and there is no difference between the “good” NAPT device and the “bad” attacker. A man-in-the-middle attacker can manipulate the packets and cause changes to the IPSec SA parameter as the NAPT does. The principle that only the initiator can change the IPSec SA parameter does not really increase the security. It leads to packet losses and complicates the protocols significantly.

- The MOBIKE is supporting the tunnel mode of IPsec only. The transport mode is not specified. The deployments can use only tunnel with ESP/AH. The ESP tunnel cannot satisfy all practical requirements, like AAA integration, forwarding performance, etc. There are stable implemented and already industry proven tunnels (like PPP), which are much more suitable for the current provider infrastructure. They work with AAA, are hardware accelerated etc. For example: PPPoE is implemented by the millions of aDSL subscribers. A new tunneling protocol requires huge investments for high scale implementations.

2.11 Related research projects for secure mobile networks

There are numerous institutes and groups working on mobility and security. The topic is not new and a lot of work has already been done by the researchers. In this thesis, we are using the experience and conclusions made in these projects. Most of them have different targets and implementation status. They are presented in the following sections.

2.11.1 Secure Mobile Architecture (SMA)

The Secure Mobile Architecture (SMA) [34] is a research project for creating a framework for secure mobility. This framework architecture enables individual design that solves specific deployment requirements. The SMA is developed by the Open Group [35], which is a vendor-neutral consortium. The main task of this group is to develop open standards between the enterprises for secure mobility and to increase the interoperability in this way. Involved in the development of SMA are: University of St. Thomas, USA [36], Boeing Technology (Boeing Company) [37], NetMotion Wireless [38] and others. There are currently test implementations at the Boeing network and no commercial deployments have been announced.

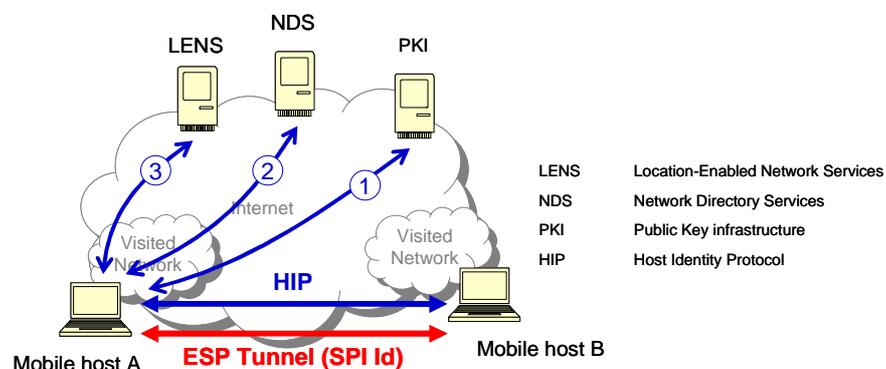


Figure 2.19: Secure Mobile Architecture

The Secure Mobile Architecture [34] delivers secure IP mobility on a high scale basis using popular protocols (Popular denotes draft or industrial not RFC protocols). There are a number of extensions defined, but the core bases on existing services. The application data are protected by IPsec protocol. The signalling is protected depending on the server but mostly SSL/TLS.

The architecture is shown in Figure 2.19. The hosts should use public/private key for authentication with x509 certificates (step 1 in Figure 2.19). The mobile host downloads the security policy and mobility information from Network Directory Services (NDS). The NDS holds the data in a virtual directory (step 2 in Figure 2.19), which contains information of the IP address and mobility policy information. The current location of the communication host

is obtained from the Location Enabled Network Service (LENS), step 3 in Figure 2.19. The LENS is a real-time location service giving the current location of the host. This type of service is typically designed for emergency calls with VoIP or location information in maps. After the policy and the current location are presented, the both hosts start negotiation directly. The Host Identity Protocol (HIP) [39] is used for session key generation, authentication and algorithm negotiation. The data are protected by the ESP protocol with HIP tag extensions described in [40].

The SMA framework requires special features at all communication participants. It is an Intranet solution where the network is controlled by a single provider (or group). Deployment in heterogeneous structure such as the Internet is not realistic, since it is not possible to implement this protocol stack in all participants. A host cannot communicate when it do not support these features. Furthermore, the NDS and LENS server are not currently available on Internet. The concept of location delivery is currently not fully clear, and thus how it can be implemented. The NAPT problem is not considered either, thus where no the direct communication is possible. There is need for specific proxies in NAPT environments.

The SMA is suitable for close Intranet environments (the Intranet must not be restricted to a small network). In this thesis, the emphasis is on open structure, which can be deployed in the heterogeneous Internet without changes on all carrier networks. The SAM require that all participating host must support mobility feature in order to communicated which is not acceptable for Mobile VPN as pointed in 2.7.

2.11.2 Enabling efficient and operational mobility in large heterogeneous IP networks (ENABLE)

The ENABLE [41] is an EU-funded research project, which concentrates on mobility in IPv6 networks. It targets an efficient mobility in large heterogeneous IPv6 networks. It will enhance the basic mobility provided by Mobile IPv6 with set of additional “premium” services, such as multi-homing, fast handovers and operator policies for service activation. The project develops long-term strategy for Mobile IP development. The project coordinates and gives input to the working groups in IETF, 3G, WWW etc. The project aims the developing real prototype, where firewalls, authentication, NAPT etc, system are working together.

Telecom Italia coordinates the project. Major partners are: Consulintel Corp.(Spain), Georg-Augus-Universität (Göttingen, Germany), Siemens AG, University of Murcia (Spain), Waterford Institute of Technology (Ireland), Brunel University (England), Huawei corporation (China).

The project has published numerous drafts in the area of application key management for Mobile IP [42], Network-based Localized Mobility Management [43], AAA with Mobile IP [44] and others. The major difference to this thesis is that this project works only on IPv6 including Mobile IP [6] and HIP [39] as a basic protocol. In this thesis, the practical solution researched includes IPv4. The IPv6 research is a long-term solution, which is not possible on the current IPv4 Internet.

2.11.3 Mobile IP group by ComNets

This is a research group on Mobile IP at department of Communication Networks (ComNets) of the University of Bremen [45], Germany. Its primary task is to research and develop advanced wireless technologies for providing continuous network access connectivity. The research focuses on (text source [45]):

- Integrated network platforms based on mobile Internet and multi-hop ad hoc protocols

- Performance evaluation and optimisation of Mobile Internet Protocols through theoretical analysis, simulation modelling and prototype development
- Determine the impact of mobile Internet protocol overheads to a range of applications such as IP-Telecommunications, FTP and the World Wide Web
- Improve the performance of the mobile Internet protocols by,
 - the elimination or reduction of protocol overheads such as Mobile Node hand-offs
 - the integration of heterogeneous wireless technologies and multi-hop ad hoc networks
 - providing enhanced terminal mobility via the concurrent utilisation of all available overlay networks (wired and wireless) in an intelligent manner that suits the users profile and the needs of the communications.

Findings and results have been published and presented at the Internet Engineering Task Force (IETF). Some of the published works are [46, 47, 48]. The research project stress ad-hoc networks and not the security of mobility IP networks.

2.11.4 Security Research Group at Microsoft Research Labs

Microsoft Research [51] labs are playing an active part in the IETF Working groups. One of the scopes of the research lab is Mobile IP Security, where the group has major influence over the Mobile IP standard [6]. Some of the publications are [49, 50]. The research group is not focusing directly on mobile VPN and it is mentioned here because of its general importance on Mobile IP.

2.12 Conclusion

Finding a secure mobile solution based on current protocols is no trivial task. The Mobile IPv4 was developed in 1996 by IETF. The protocol was rewritten in 2002, the success of Mobile IPv4 seems only to be at research level and partially in CDMA 2000. Its successor, Mobile IPv6 improves the protocol properties significantly and it will be part of all IPv6 implementation. The Mobile IPv6 will be part of the future IPv6 backbone, but there are less ISP deployments currently.

The first solution candidate is using two-layer principle - one layer for mobility and one layer for security. In the 2.10.1 is clearly shown that this type of solution has design problem. The mobile signalling cannot be protected since the security layer cannot be under the mobility. The mobility layer provides the static layer for the security session. Furthermore, Mobile IPv4 is not compatible to NAPT. The two-layer solution cannot principally fulfil the protection of the mobile signalling.

A different approach is the one layer solution. The best representative is the MOBIKE protocol based on IKEv2, see 2.10.2. The solution combines security and mobility in one to achieve mobile security. The protocol is a huge step to the necessary solution, but suffers from practical disadvantages. The deficits are a consequence of the static IKEv2 protocol see 2.10.2.3.

The use of IPSec as a protection makes the session traceable. A man-in-the middle attacker can trace physical movements. Even without knowing the exact used ID in the session, this is a security problem.

All existing candidates for secure mobility have a lack of efficient location update mechanism (called natt-keep-alive, dead-peer-detection, status check or path testing). The updates are suboptimal regarding disconnection and network load. An efficient mechanism executes updates depending on the movements of the Mobile Node and the network status see 2.6. The target is, on the one hand, to reduce the update number to save network resources and on the other hand to minimize the disconnection time. This can be achieved by smart distribution of the updates as shown in this thesis see chapter 7.

2.13 References in chapter 2

- [1] Alvarez, S., "QoS for IP/MPLS Networks", Macmillan Technical Publishing, Juni 2006
- [2] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002
- [3] Tanase, M., "IP Spoofing: An Introduction", <http://www.securityfocus.com>, 2003
- [4] Perkins, C., "IP Mobility Support for IPv4", RFC 3344, August 2002
- [5] Deering, S., R. Hinden, Editors, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, 1998.
- [6] Johnson, D et al., "Mobility Support in IPv6", RFC 3775, June 2004
- [7] Montenegro, G., "Reverse Tunneling for Mobile IP (revised)", RFC 3024, January 2001
- [8] Perkins, C., "IP Encapsulation within IP", RFC 2003, October 1996.
- [9] Adrangi, F; Levkowitz, H "MIPv4 VPN Traversal Problem Statement", RFC 4093, August 2005
- [10] Devarapalli, et al., "Network Mobility (NEMO) Basic Support Protocol", RFC 3963, January 2005
- [11] Soliman H. et al. "RFC 4140 - Hierarchical Mobile IPv6 Mobility Management (HMIPv6)", RFC 4140, August 2005
- [12] Kent, S., and R. Atkinson, "IP Authentication Header", RFC 4302, December 2005
- [13] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005
- [14] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.
- [15] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.
- [16] Huang, et al., "A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers", RFC 3706, February 2004
- [17] Piper, D., "The Internet IP Security Domain of Interpretation for ISAKMP", RFC 2407, November 1998.
- [18] Narten, T., Nordmark, E. and W. Simpson "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.
- [19] Thomson, S. and T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC 2462, December 1998.
- [20] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C. and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.
- [21] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, December 1998.
- [22] Arkko, J., Devarapalli, V., and F. Dupont, "Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents", RFC 3776, June 2004.
- [23] Kaufman, C., Ed., "The Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.
- [24] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.
- [25] Kent, S., "IP Authentication Header", RFC 4302, December 2005.
- [26] "Cdma2000 Wireless IP Network Standard: Simple IP and Mobile IP Services", 3GPP2, X.S0011-002-C v2.0, June 2005
- [27] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.
- [28] Eronen, P., "IKEv2 Mobility and Multihoming Protocol (MOBIKE)", RFC 4555, June 2006.
- [29] Kivinen T, Tschofenig H., "Design of the IKEv2 Mobility and Multihoming (MOBIKE) Protocol", RFC 4621, August 2006
- [30] The Internet Engineering Task Force (IETF) www.ietf.org
- [31] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowitz, "Extensible Authentication Protocol (EAP)", RFC 3748, June 2004.
- [32] Rivest, R., Shamir, A., and Adleman, L., "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM, v. 21, n. 2, February 1978.
- [33] Jonsson, J. and B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", RFC 3447, February 2003.
- [34] The Open Group, "Secure Mobile Architecture (SMA) Vision and Architecture", Technical Study, February 2004
- [35] The Open Group, www.opengroup.org
- [36] <http://www.stthomas.edu/engineering>
- [37] <http://www.boeing.com/commercial/techsvcs/boeingtech/index.html>
- [38] <http://www.netmotionwireless.com>
- [39] Moskowitz, R. and P. Nikander, "Host Identity Protocol (HIP) Architecture", RFC 4423, May 2006.
- [40] Jokela, P., "Using ESP transport format with HIP", draft-ietf-hip-esp-05 (work in progress), Feb 2007
- [41] ENABLE Project, www.ist-enable.org
- [42] Giaretta, et al., "Application Master Session Key (AMSK) for Mobile IPv6", Mobile IP v6 Working Group , October 2006
- [43] J. Kempf, Editor, "Goals for Network-based Localized Mobility Management (NETLMM)", IETF draft, October, 2006
- [44] Giaretta, et al. "AAA Goals for Mobile IPv6 ", IETF Draft, September 2006
- [45] Mobile IP by ComNets, <http://www.mobileip.org/>
- [46] Aust, S., et al. "Hierarchical Mobile IP ns-2 Extensions for Mobile Ad hoc Networks", 4th IASTED International Multi-Conference on Wireless Networks and Emerging Technologies (WNET 2004)Banff, Alberta, Canada, July 2004.
- [47] Aust S. et al. , "Proactive Handover Decision for Mobile IP based on Link Layer Information", First IFIP International Conference on Wireless and Optical Communications Networks (WOCN 2004)Muscat, Oman, June 2004.
- [48] Fikouras N., Kuladinithi K. et al., "Multiple Access Interface Management and Flow Mobility", 13th IST Mobile and Wireless Communications Summit 2004, Lyon, France, June 2004.
- [49] Arkko, A., "MIPv6 BU Attacks and Defenses", IETF draft, February 2002

-
- [50] Roe M., Aura, T., O'Shea G., "Authentication of Mobile IPv6 Binding Updates and Acknowledgments", IETF draft February 2002
 - [51] Security Research Group, <http://research.microsoft.com/security/>
 - [52] Gundavelli, S., et.al, "Proxy Mobile IPv6", RFC 5213, November 2007.
 - [53] Wakikawa, R., et al. "IPv4 Support for Proxy Mobile IPv6", IETF Internet-Draft, April 2009
 - [54] Huttunen, A., et al. "UDP Encapsulation of IPsec ESP Packets", RFC 3948, January 2005.

3 Mobile Virtual Private Network

Mobile Virtual Private Network (Mobile VPN or M-VPN) is a new protocol delivering secure IP mobility. It introduces new features and covers the opened issues described in chapter 2. The M-VPN consists of three sub protocols, which together deliver secure IP mobility:

- Mobile Key Exchange (M-KE) is a protocol for the negotiation of session parameters in mobile environments. The peers are identified, authenticated and authorized during the negotiation. The protocol delivers crypto algorithms, session key, session ID etc, for the following protection protocol M-SE. The protocol operates over IP/UDP. Its structure is described in 3.5 and it is fully specified in chapter 4.
- Mobile Secure Encapsulation (M-SE) protects the communication in mobile environments. The protocol secures the application data and mobile signalling through encryption, authentication, integrity check and replay protection. The M-SE has an anti-tracing mechanism, thus the session cannot be matched to IP addresses. The protocol works on an application layer and delivers transparent transport layer. The applications can use the transport layer (TCP/UDP) directly or can implement an IP tunnel. The protocol itself does not have tunnel properties. It integrates existing tunnel protocols, like L2TP or GRE. The structure of M-SE is described in 3.6 to 3.11. Chapter 5 gives the technical specification.
- Mobile Location Update (M-LU) is a mathematical algorithms description rather than a technical specification. The protocol optimises the update intervals regarding minimal disconnection and resources. This is a very important requirement for the mobility. The structure of the M-LU is presented in chapter 7. The M-LU is derived in chapters 8 to 10.

3.1 Principles of Mobile VPN

The IP header currently has dual purpose for host identification and network routing. The PoA parameters cannot be used for the identification in the mobile environments (def: PoA - IP, TCP, UDP, ESP etc.). The network parameters can frequently change during one mobile session because of change of the access network.

The dual purpose is split between PoA and M-SE. The PoA parameters are used only for transport to the destination. The packet is identified at the M-SE layer independent from the PoA parameters. The main principle is that if the packet is authenticated, then it must be assigned to the right session regardless of the network or transport parameter. The M-KE and M-SE do not rely on the identification of under layers, i.e. IP or UDP parameter.

The PoA parameters can be replaced by intermediate NAPT devices or by an attacker. It is not possible to differentiate between “good” and “bad” header manipulations. For this reason, the PoA parameters are considered untrustworthy. If the IP and UDP are not trustful, then their influence on the security layer must be considered carefully. For example: ICMP notification for path MTU discovering (see A.2.2.7) [2] or host unreachable [3] must be carefully inspected and their influence on the security reduced. The restriction must be done using the local policy created and defined by the administrator (see 6.2).

The NAPT devices are used in most of the Internet access networks as shown in A.2.1. A protocol friendly to NAPT must in general follow the client-server architectures at network

layer. Peer-to-peer at network layer is not possible behind dynamic NAT router, because of the unidirectional establishment of the session. In protocols like SIP [4] the NAT was not considered by the protocol design, which led to dramatic problem when implementing the security [6].

It must be pointed out that peer-to-peer architecture depends on the abstraction level. It is possible to develop peer-to-peer topology at higher layer, where on network layer is user client-server architecture. In fact, most of the peer-to-peer networks (file sharing) have client-server architecture at the network layer (Skype, DC++ etc.). M-VPN implements client-server model at a network layer. It can deliver peer-to-peer abstracted at the higher application layer. Peer-to-peer topologies are described in 3.3.

The M-VPN must achieve maximum effect with minimum resources. The protocols must be simple to implement and based on existing crypto libraries. Complex protocols are difficult to understand and there is a higher risk of misunderstanding or wrong implementations. A modern protocol must integrate existing stable libraries, so that fast implementation is enabled. The use of common libraries increases the interoperability.

The protocol is not starting on a green field. The companies and Service Providers have already network infrastructure and use already authentication and authorisation servers. It is not realistic to consider that a protocol can succeed, when it does not fit in the established networks types and AAA processes.

3.2 Targets of Mobile VPN

The protocol concentrates on delivering secure IP mobility to the current Internet using the gathered experience on the existing protocols. The technical targets are:

- Protecting the mobile signalling and application data with encryption, authentication, integrity, and replay protections as defined in 2.6.
- The location updates must be proportional to the mobile host movements in order to reduce the host and network resources. More movements are reflected in more updates and vice versa.
- The protocol assumes that the mobile host is not aware of its PoA parameter. (see 2.4 and 2.5). The mobile host may use mobile gateway to be informed for its PoA. The procedure is denoted as location update in 2.2.
- The M-VPN protocol must use IP and UDP for transport. The use of TCP is not possible, since it requires static IP addresses. Furthermore, the use of TCP for transport causes serious problem by packet losses as described in 3.13.
- The protocol must build a client-server model. A peer-to-peer connection could be built at higher abstraction layer.
- The M-VPN protocols must itself implement datagram fragmentation in difference to IKE and ESP. The reason for fragmentation is that the network layer does not have the capacity to transport the data in one packet. In this case, the data is split and transmitted in multiple fragments. M-VPN is running over UDP and according the RFC 768 [9] fragmentation must be done at the IP packet. The IP fragments may be dropped by the NAT devices if they are received out of sequence, as described in A.2.1.9 [11] (Firewalls have the same behaviour). The firewalls also treat out of sequence fragments as potential attack. In wireless networks (3G, WiFi), it is normal to receive out of sequence IP packets. To overcome this situation, fragmentation in the M-VPN protocols must be done and the UDP header left complete in

all packets. To increase the performance, the fragmentation must be carried out only in M-KE and as rarely as possible.

- The protocols must be able to work in fast changing networks. Circular dependencies can occur in this type of networks, when the inner processing time of the packets is longer than the interval between the host movements. For example: when authentication request is processed by the server for 20 sec and the Mobile Client is moving every 10 sec. During the processing of the message, the transport parameters alter. The response packet cannot be sent to the current PoA of the mobile host, since it has moved to a new one. To avoid circular dependences, polling and caching mechanisms must be implemented in the M-KE protocols. The caching delivers the ability for fast processing of repeated requests (see 3.5).
- The protocol should integrate the existing cryptographic algorithms. Fast deployment due to existing libraries and high security due to already deeply studied properties, can be delivered in this way.
- Already implemented authentication mechanisms must be supported. For example EAP [7] and RSA [8] signatures are widespread and there is no reason to replace them.
- The protocols must support the existing authentication, authorisation and accounting structures, like Radius servers.
- The M-VPN protocol should not have tunnel features. These should be delivered by well-established protocols, such as L2TP (PPP) [13]. The creation of a new tunnel protocol is not very suitable, because it requires new authentication and new hardware acceleration cards (see 3.3.1). Another major advantage is the ability to separate Tunnel Node and the Mobile Node at different hosts (see 3.3). The tunnel protocols will typically deliver IP connectivity, but it could also deliver constant link layer.
- The packets to the MN are routed in a tunnel interface using standard routing. Interception, as in Mobile IPv4, can cause problems with the link layer security infrastructure. The actions of the Home Agent (HA) in Mobile IPv4 can be treated as an attack, since the HA hijacks the ARP [7] mapping of IP to Ethernet MAC.
- To facilitate the assigning of IP parameters, the tunnel must be implemented as virtual interface. (IP parameters can be assigned only to an interface)
- The protocols should be carrier grade in the sense of delivering a pan regional backup possibility and the ability for load sharing.
- The new applications are required only at the end hosts, thus Mobile and Server Node (see 3.3). The protocol must be transparent for the intermediate Internet devices, like routers, firewalls etc.
- Route optimisation should be supported. It is not a primary target of the M-VPN protocol, since it is not always possible in NAPT networks as described in 3.2.1.
- Bundled session must be avoided, since they have problem with NAPT devices, see A.2.1.5.
- The protocols must support IPv4 and IPv6.

These technical requirements are result of the experience on IPSec and Mobile IP and target easy and efficient integration.

3.2.1 Route optimisation in Internet

Route optimisation means direct communication between the hosts using the shortest path from a routing perspective. Proxies break the direct communication, because the packets are routed first to the proxy and then to the destination host. This leads to suboptimal routing, which increases the communication delay. An example is a host in Germany using a proxy in Japan to browse web pages in France. The packets are first routed from the German host to the proxy in Japan and then are routed back from Japan to the web server in France. The delay will be significantly bigger then a direct communication from the German client to the web-server in France.

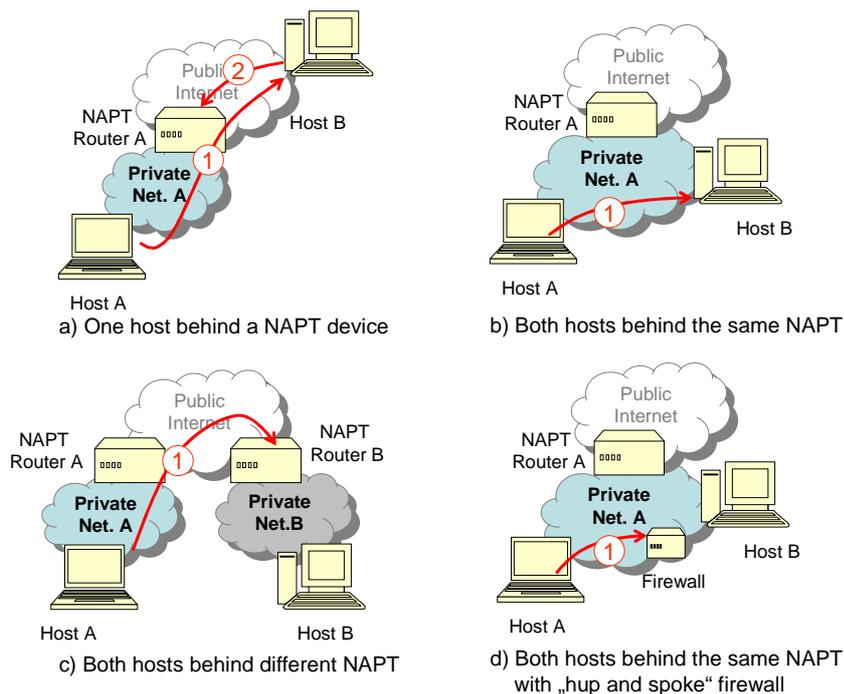


Figure 3.1: Route optimisation and NATP

The route optimisation is a big advantage of the IP network. Unfortunately, this is not always possible in NATP environment (see A.2.1). The dynamic NATP allows the connection to be initiated from inside only. Direct connection in the routing sense is not always possible. There are five different constellations:

- Only one of the hosts is behind a NATP device, Figure 3.1 a). In this case, routing optimisation is possible, but the connection must be established from inside the NATP, step 1. The connection from out side would fail, see step 2 in the figure.
- Both hosts are behind the same NATP device shown in the Figure 3.1 b), for example the same WiFi network. Route optimisation can be used if the communication between the hosts is not restricted by a firewall.
- Both hosts are behind different NATP devices, shown in Figure 3.1 c). This is a very common scenario, when the host hosts access Internet on different physical location. For example: WiFi hotspots in Munich and Frankfurt. None of the host can establish direct connection to the other one, because

both of them are behind of NAPT router. Route optimisation is not possible and the hosts must communicate through a proxy.

- Both hosts can be behind the same NAPT device, but the communication between the hosts is restricted by firewall (shown in Figure 3.1 d). Most of hotspot providers restrict the communication between the hosts in the LAN (“hub-and-spoke”). Even in the same network, the hosts cannot communicate directly with each other. This restriction at the local firewall is done for security reasons (protection against local LAN attacks). Although it is very controversial, the policy is in place.
- There is no NAPT between the devices. This is the best case, where route optimisation is certainly possible.

The route optimisation is not possible in many real cases. It is even difficult for the hosts to determine the exact NAPT constellation. The private IPs used behind the NAPT usually overlap and therefore, the knowledge of the IP address is insufficient to determine if the hosts are in the same network. Trying to exchange packets is the only way to find out if direct communication is possible. Sending packets and waiting to see if there is an answer, is not ideal for real time applications. There are protocols, like ICE [12], which try interactive to established direct connection when possible. The protocol is an extension of SIP [4] and controls the target IP addresses in SDP. The experience of ICE protocol shows that the benefits of trying on-the-fly to optimise the connection are far less than the implementation complexity. There are more states and error combinations.

Interactive re-routing to a shorter path is a security issue too. A possible attacker can try to hijack a connection by pretending to be closer to the destination host. The security structure and states become more complicated and the risk of implementation defects increases.

The network topology in mobile environments changes frequently because of host’s movements. The topology can change from transparent to behind different NAPT routers, thus from Figure 3.1 b) to c). Monitoring and switching during the communication require many resources on the mobile hosts. The communication session becomes very instable due to packet losses.

Mobile VPN should deliver mechanism for routing optimisation, but its implementation is optional. The route optimisation makes more sense in closed networks, like provider’s backbones and Intranets.

3.3 Architecture overview of Mobile VPN

The M-VPN defines host types and network element. The terminology must not be confused with Mobile IP or IPSec although there are some relations. The roles of the nodes are different in Mobile VPN since the target and architecture is different. To prevent the reader from making some wrong association to the existing protocols, the nodes roles are defined:

- *Mobile Client (MC)* is a host moving through different networks, changing its IP address. A typical example is a small hardware device, like a smart phone attaching to different IP networks whilst communicating. The M-VPN delivers a static IP layer to the applications running on the Mobile Client. The mobile host establishes a connection to the Mobile Server, thus the mobile host is the initiator of M-KE and M-SE connection.
- The *Mobile Server (MS)* accepts connections from a Mobile Clients. The Mobile Server must not have static IP address and includes mobile

functionality too. In contrast to the Mobile Client, the Mobile Server cannot initiate connection. The Server can be only a responder. Furthermore, the server can provide information for other Mobile Servers in the same administrative domain. In this way, dynamic overlay networks can be created. It is subjectively assumed that the server is moving more slowly than the Mobile Node. It is assumed that the Mobile Client is aware of the Mobile Server's IP even if it changes.

- *Mobile Node* (MN) is a host being either Mobile Client or Mobile Server. The term is used to describe properties typically for both: Mobile Client and Server. The term is more general than Mobile Node in Mobile IP.
- *Tunnel Node* (TN) provides tunnelling functionalities. It creates and terminates tunnels. It could be standard L2TP LNS/LAC or GRE. The Tunnel Node is an optional element, which is external to M-VPN protocol. The node does not need any specific M-VPN functionalities but it is relevant for the protocol.
- *Home IP parameter*. The constant IP parameters assigned by the tunnel node (TN) are called Home IP parameters. These are used by the applications. They remain permanent during the PoA changes (moving to different access IP networks).
- *Home Network* is called the network corresponding to the Home IP parameters.

Other elements (nodes) relevant to the protocols and already existing parts of the network infrastructure are *AAA servers* (Authentication, Authorisation and Accounting). They are usually part of the Intranet infrastructure and mostly use protocols, like Radius [23], Kerberos [24] etc.

The M-SE protocol delivers transparent transport layer and constant network layer. The constant network layer is not transparent because the IP addresses are exchanged by M-SE with IP Mappers, see 3.6.1. Over the transport layer, a tunnel can be built by the Tunnel Node (TN), which is independent of the M-VPN. This is called *tunnel mode* of operation. The TN must be present on both sides (start and termination point), thus Mobile Client and Mobile Server. The TN can be deployed physically on the Mobile Node (MN) or on a separate host. The TN is usually part of the Mobile Client (MC), such as L2TP LAC. In contrary, the TN will be typically implemented on a different host at the Mobile Server's (MS) side. The MS is typically an aggregation point of multiple M-SE sessions and the separation increases the performance of node. To emphasize this possible implementation, the MC is drawn together with the tunnel end point in Figure 3.2, Figure 3.3 and Figure 3.4. The MS and its TN are drawn as two separate logical elements.

The M-VPN can be also used in so-called *native mode*, see 3.6. There is not Tunnel Node in this deployment. The native mode is much more restrictive and can be used in fewer scenarios. The native mode is not the main target of the M-VPN and not described further in this section.

There are principally two types of usage of M-VPN in tunnel mode: Intranet and Internet usage. In the first case, the MC accesses the Intranet, such as corporate LAN. This is shown in Figure 3.2. The home network is part of the Intranet. The M-SE session protects the exchanges for achieving mobility. A tunnel such as PPTP, L2TP [13] is built over the M-SE session. The TN is called LNS in the L2TP terminology [13]. The MC has two IP addresses, one corporate (typically private) and one public IP. The corporate IP is permanent and used for the communication. The public IP is dynamically changing according to the MC

movements. The communication hosts to the Mobile Client must also be part of the Intranet. The Mobile Client cannot communicate with its corporate IP to Internet.

In the Internet usage, the home network consists of public routable addresses as shown in Figure 3.3. The MC has two public IP addresses. The first one is a constant IP of the tunnel interface. The second IP is dynamic and used for Internet access. The Mobile Client can communicate to all Internet hosts using its public Home IP and vice versa.

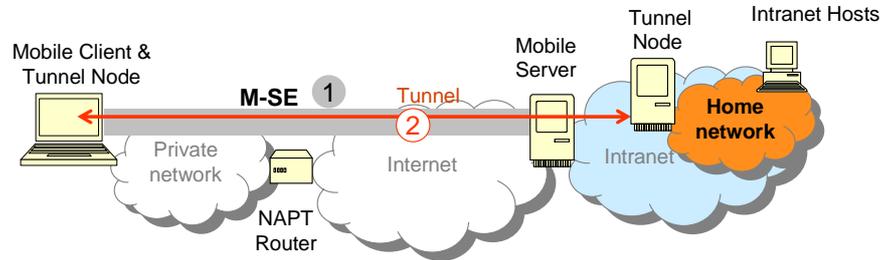


Figure 3.2: M-VPN for Intranet access

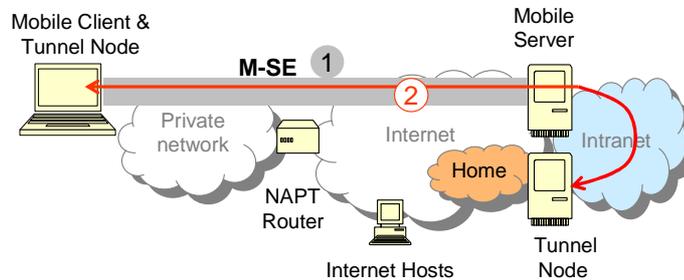


Figure 3.3: M-VPN for Internet access

3.3.1 Advantages of the physical separation between Mobile Node and Tunnel Node

The advantages of using external Tunnel Node are:

- There are already developed Tunnel Nodes, which can serve thousands of parallel sessions using hardware acceleration modules. They are a well-studied and stable implementation. For example: all aDSL customers use Point-to-Point Protocol (PPP) and the ISPs server millions of customers.
- The existing TNs are already integrated into AAA structures.
- There are synergy effects of using the same infrastructure for M-VPN, PPPoE in aDSL, L2TP over IPsec etc.
- The existing TNs integrate typically dynamic routing protocols, like ISIS and BGP, which is major requirement of the Network Service Providers.
- A new Tunnel Node in M-VPN requires new hardware implementation, new settings at AAA etc. This will complicate significantly the deployment.

3.3.2 Redundancy and load sharing in M-VPN

The M-VPN enables working in topologies with parallel connections to the same resources. This improves the availability through redundancy. The Figure 3.4 presents redundant connection, where the same home network is reachable through two separate MSs. The M-VPN delivers load sharing when the redundant connections are actively used. The load sharing is a routing problem. In M-VPN, this is left to the standard routing mechanism. Generally, the routers perform load sharing, when there are multiple paths with the same

metrics for the same destination. The two connections in Figure 3.4 can be used for load sharing and redundancy. The routing protocols can run statically or dynamically over the tunnel.

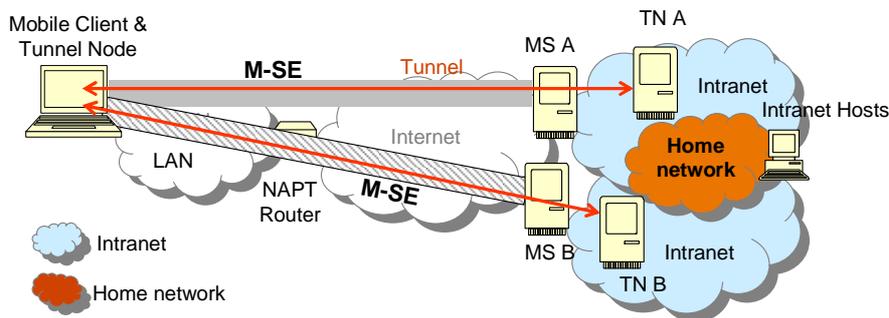


Figure 3.4: Redundant connection with M-VPN

3.3.3 Dynamic overlay network with M-VPN

M-VPN breaks the static topologies where the peers must be predefined. The Mobile Nodes can discover resources and establish new connection. A dynamic overlay topology can be build using this property. Dynamic network topology is very important because in mobile environment the optimal provider to resources may dynamically change. This improves significantly the recovery time by network failures. Furthermore, it enables path diversity in VPN and load sharing.

The key feature is a procedure, called Mobile Server discovery (see 3.8 and 4.10.1). The Mobile Server provides information to the Mobile Clients for other known Mobile Servers and optional their resources, like Web services, Videos etc. The procedure is part of M-KE protocol. The Mobile Client requests resources information and the server responds with a list of known Mobile Server. The server list contains objects representing resources, like IP addresses, home parameter, ID, media content etc. The criteria for establishing a connection to certain Mobile Servers depends on the Mobile Client policy and can be every parameter of the server list, such as the ID, round trip delay to the destination, etc.

The Mobile Servers provide the list to the Mobile Clients. The Mobile Clients cannot announce a list to the Mobile Server in opposite way. The Mobile Client may only nonce single resource running at the same host. The same host can run Mobile Client and Server Client, thus the host can accept connection and initiate connection at the same time. In this case, the Mobile Node can announce to other servers that it is also running a Mobile Server. The announced resources are added in the server list of the other Mobile Server. The Mobile Node cannot announce resources, which are not directly provided by him. This protects from avalanche effect, when every Mobile Node announces all known elements. The system will be overloaded because the same resources will be re-announced permanently. This simple rule that only directly provided resources can be announced, assurances single object per resource in the list.

Every Mobile Server has its own list of other resources. There is no general master list between all nodes in Mobile VPN, which requires version managements and time synchronisation etc. There is no desire to rebuild the existing dynamic routing protocols, where the target is to maintain a consistent routing table over multiple elements. The routing protocols may be used between the Mobile Nodes implemented in classical way. There are no particular restrictions because of M-VPN, thus there is IP connectivity between hosts.

An overlay network built with Mobile VPN is presented in Figure 3.5: Dynamic overlay network. The Mobile Client A can reach the node G via two different paths: A-B-D-G or A-

C-D-G. Which of them is better and preferred can be determined with the routing protocols running on the top Mobile VPN. The different routing protocols have different convergence time. The convergence time is the time for updating the routing entries, so that the routing table at every router is consistent. Protocol with fast convergence is ISIS [17, 18] and with slow can be considered RIP [19].

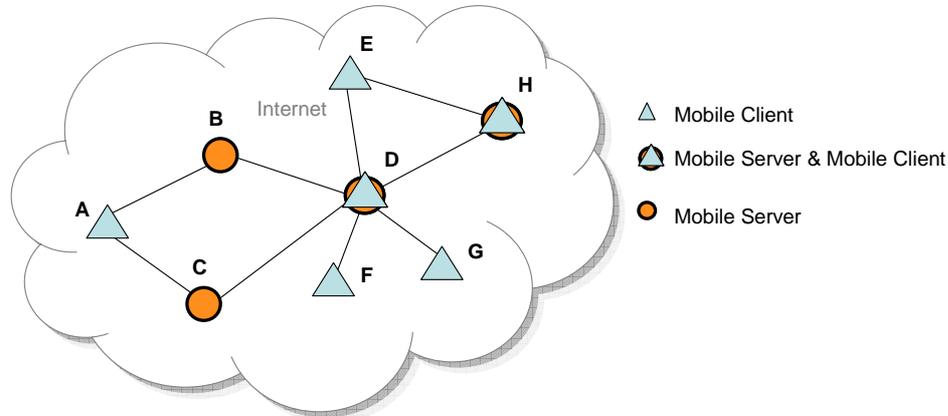


Figure 3.5: Dynamic overlay network

The Mobile VPN offers mechanisms for building dynamic meshed networks in a mobile and secure way. The transparent bidirectional connectivity in Internet can be restricted by NAPT as already mentioned in 3.2.1. The NAPT reduces the mesh possibilities of M-VPN too.

3.4 Bundling of M-KE and M-SE

One major requirement of M-VPN is to work in NAPT environment. Bundled sessions have a big problem in NAPT as described in A.2.1.5 because the payload must be properly interpreted by the NAPT router. The bundled sessions mean that the application uses multiple sessions at transport layer. A simple example is the FTP protocol (active mode). There are one control TCP session and multiple data TCP sessions. The control TCP session is established ingress (from Client to Server) direction and data egress (from Server to Client) direction. Both sessions have different source and destination ports. The NAPT must recognise the dependency between the sessions and correctly predefine bindings. The same problem exists in VoIP for example, where a signalling and data are different UDP sessions. The term session must be defined even for the connectionless UDP protocol in order to maintain the NAPT binding (see A.2.1).

There are two approaches to solve the NAPT issue. (1) The first one is to make the NAPT device aware of the dependencies between the sessions in the protocols. This is done by the manufactures for FTP, VoIP etc. (2) The second approach is to change the protocols in a way, that there is a single session at network layer or a session established at the same direction. There is no need of any change at the NAPT device. This is a clear target for the Mobile VPN. The same NAPT binding must be used for M-KE and M-SE between the hosts. To match the same NAPT binding, the protocols use the same source and destination IPs and ports.

The Mobile Node (MN) will receive UDP messages on the same port from M-KE and M-SE. Therefore, the MN must implement two-level asynchronous packets handling shown in Figure 3.6. The first process binds on the socket and distributes the packets between the M-KE and M-SE processes at the second level. Fast dedifferentiation between the M-KE and M-SE packet is achieved by comparing the first byte of the message. The first byte is the protocols version. M-KE has possible version 0 to 127 (8 major and 16 minor). The M-SE

uses versions 128 to 255 (8 major and 16 minor). Only by inspecting the first byte, the socket daemon (process) can find out the protocol. The daemon must also FIFO buffer the packets for sending and receiving. At the second level are the processes of M-KE and M-SE, which asynchronously read from the buffer and handle the packets.

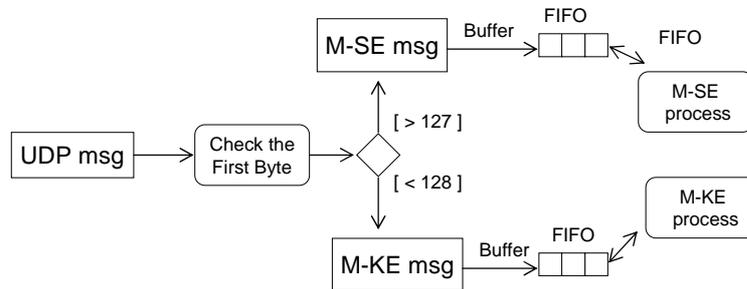


Figure 3.6: Processing of M-KE and M-SE messages

3.5 M-KE overview

The Mobile Key Exchange (M-KE) has the following targets:

- Operation in mobile environment with frequent changes of IP.
- Identification and authentication of participants using a variety of algorithms, like digital signature with x509 certificates, shared secret, password etc. The protocol must integrate in the current AAA structures.
- Negotiation of cryptographic parameters for M-SE session, like encryption, authentication, integrity protection etc.
- Delivery of authenticated key material for M-SE session, thus session keys for encryption, authentication and integrity protection.
- Announcement of the mobile parameter of the nodes, like minimal interval supported between the PoA changes.
- Exchange of network and service information. They describe the network or services directly connected to the server and client. Networks connected by hops (not directly connected) are not considered.
- The protocol allows information's exchange about other known servers offering resources. The nodes can build overlay networks with dynamic topologies in this way.

The M-KE runs over UDP and is fully defined in chapter 4. The Mobile Client initialises the negotiation using the pre-configured Mobile Security Policy (M-SP). The Mobile Server is always the responder and cannot initiate a session.

The M-KE delivers the parameters for M-SE after a successful negotiation. The M-KE session is deleted after the negotiation, thus there is no permanent session, like in IKE (s. A.2.2). One M-KE session results in one M-SE session. If there is a need for more than one M-SE session, then the M-KE must be repeated. A single M-KE for building multiple M-SE sessions makes sense from theoretical perspective, but it has many practical disadvantages, like permanent session, need for keep-a-live mechanisms, out of the state by network failures.

3.5.1 Mobility during the M-KE

In order to define how fast can change the PoA, the term mobility in negotiation must be defined. A negotiation in total mobility cannot be designed principally. There is a circular

dependency if the time between movements is less than the packet transmission time. The IP/port of received packet is already out-of-date, since the sender already has a new one. The nodes can principally communicate only if the host has a constant PoA for at least the round trip delay plus the local processing time. The negotiation cannot be achieved if one of the hosts is moving faster than this interval. This is not a restriction specific to M-KE but one that hold for any other protocol as well. The IP and port can change during the negation but the change must be bigger than sum of round trip time with the processing at host. The requirement can be mathematically expressed as:

$$\begin{aligned} t_{constPoA}^{client} &\geq t_{send} + t_{receive} + t_{proc}^{client} \\ &\geq 2 \cdot t_{send} + t_{proc}^{client} \\ t_{constPoA}^{server} &\geq t_{send} + t_{receive} + t_{proc}^{server} \\ &\geq 2 \cdot t_{send} + t_{proc}^{server} \end{aligned} ,$$

where $t_{constPoA}^{client/server}$ denotes the time for which the client or server must have constant PoA. The $t_{send/receive}$ denotes the time for sending or receiving a packet. The time for sending and receiving could be assumed as equal (this is in most of the cases). Potentially, the asymmetric DSL bandwidth can lead to different delays but this is out of scope in this section. The $t_{proc}^{client/server}$ is the time for packet processing, thus the time needed to generate an answer of the received packet. The processing time includes not only the physical processing time dependent on the gateway's CPU but also the AAA authentication by remote server, thus looking in remote databases etc. It should not be underestimated, since by roaming for example, the authentication is made by remote system and it can take up to a minute. Furthermore, involving searches in databases leads to processing time of several seconds.

3.5.2 Polling and caching in M-KE

To increase the mobility, the interval with constant PoA must be reduced. The transmission time cannot be influenced because it is a physical property. The only way to reduce the interval with constant PoA is working on the processing time at the MC (Mobile Client) and MS (Mobile Server). The key idea is to split the negotiation in multiple parts, between which the PoA may change. The polling mechanism reduces the requirement on the processing time, thus improves the mobility.

The M-KE exchange consists of multiple request and response messages. Mobile Client sends the request and the Mobile Server sends the response. A pair of request and response is denoted as *mobile exchange pair*. The M-KE consists of multiple mobile exchange pairs.

The key requirement is that the Mobile Server must send the response message immediately to Mobile Client, thus with out any delay. There two types of possible responses: pending and response data. The pending is send when the server cannot process the request immediately. For example, if the Mobile Server needs to perform DB search taking around a second, then it responds with pending. The pending message contains the time in seconds, when the Mobile Client must repeat the request. If the Mobile Server can answer immediately, then it sends the data to the PoA of the request message. The PoA of the Mobile Client may change between the mobile exchange pairs. The Mobile Server uses always the PoA of the last request message of the Mobile Client.

Every mobile exchange pair can be repeated several times until successful exchange. An example is shown in Figure 3.7. The client sends its authentication credentials, like username and challenge response. The server cannot verify the credential immediately and therefore answers with pending notification to the PoA of the request. The polling interval in the server

response is 50 seconds. The MC waits for 50 second and resends the authentication packet for the second time. The PoA in the retransmitted M-KE packet can be different that the first one, thus between the exchanges pairs the PoA can change. The server is still not ready with the authentication and authorisation and again answers with pending notification. The polling interval is 20 seconds. The client repeats the request in 20 sec from different PoA. The server has finished the authentication meanwhile and answers immediately with successful

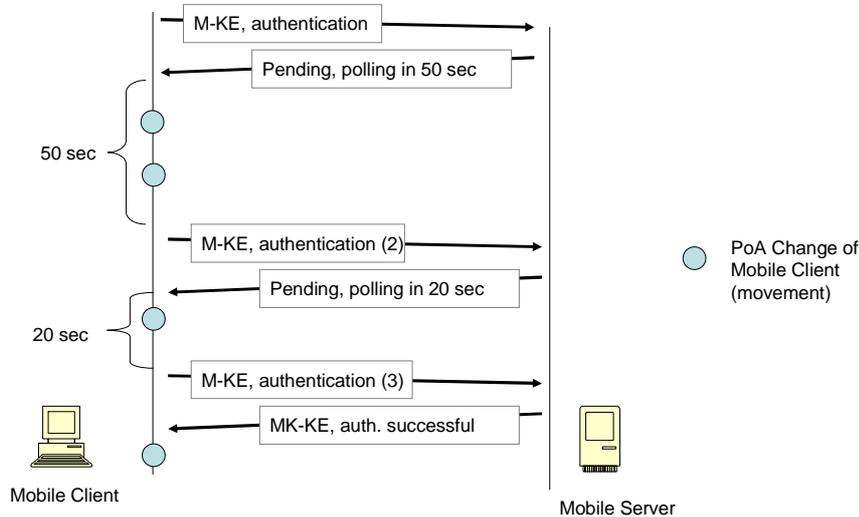


Figure 3.7: M-KE polling example

The server must implement caching to fulfil the polling mechanism. It must be able to keep client information in the memory, so it can be replied immediately. This is mostly relevant for the authentication and the authorisation, where the AAA server may be in multiple hops distance. The AAA answer can take many seconds. The current Radius servers do not use validity period, thus the answer is valid only at the time point of the response. In M-KE, validity of the AAA answer must be set. It is called caching period. The caching period must be at least the polling interval plus the $t_{constPoA}^{client/server}$. Otherwise, the AAA response will expire between the pollings. The challenge and the response values must be cached when working with passwords. Caching password means that the same challenge must be involved, because the password can be verified only through the challenge value.

3.6 M-SE overview

The M-SE is an application protocol over UDP fully defined in chapter 5. The M-SE parameters can be classified in three groups: security parameters, mobility parameter and data parameters. The security parameters are keys, algorithms for the protection etc. The mobility parameters consider the mobility properties of the Mobile Nodes, like the minimum PoA change interval supported by the Mobile Server. They give common estimation of the node performance and the network quality. The data parameters are traffic selectors defining which packet must be protected by the M-SE session. The traffic selectors are pre-defined at the Mobile Nodes. They are announced during M-KE and the biggest common denominator is set. The selector defines the transport layer parameters such as protocol and ports.

The M-SE achieves confidentiality through encryption, authentication and integrity through HMAC [32], reply protection. The protocols deploys anti tracing mechanism by using pseudo random values in the header (see 3.7). The M-SE delivers transparent transport layers to the server. Transparent means that it is not manipulated or intercepted between the end-hosts. The network layer (IP) is constant, but not transparent, thus IP's are replaced between the ends.

The protocol operates in *tunnel* and *native mode* (see 3.3). The tunnels, like L2TP and GRE can be built between a Tunnel Node (TN) and Mobile Node (MN). In native mode, the communication between single applications is achieved. The layers in tunnel and native modes are shown in Figure 3.8.

The tunnel is built on top of the M-SE layer. The Tunnel Nodes (TNs) administrate Home IP parameters, which are unknown to the Mobile Nodes (MNs). The TNs can authenticate and authorize additionally to M-SE. This is common to the L2TP (PPP) implementations. The authentication and authorization duties can be distributed between the TN and MN according 6.1. The Figure 3.8 a) presents the integration with the two main tunnel types, i.e. L2TP and GRE. The use of tunnel requires more resources, but it is more flexible. It is recommended over the native mode.

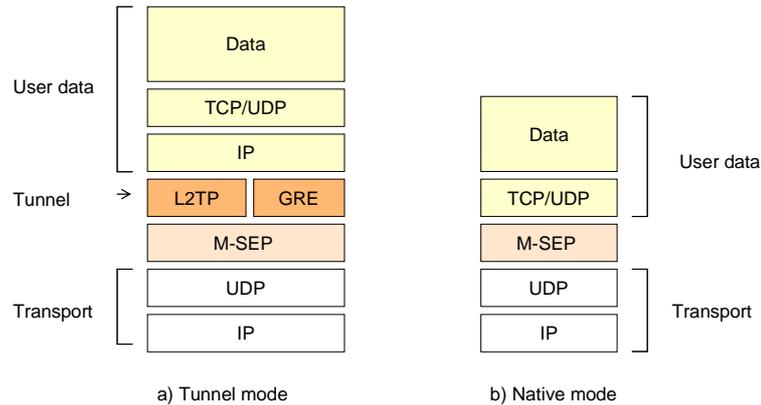


Figure 3.8: M-SE mode of operation

M-SE in native mode protects the communication between two predefined application hosts. The Mobile Node and the application host can be the same or different physical node. The applications can use TCP or UDP for establishing the connection. The layers are shown in Figure 3.8 b). This form of protection must not be confused with TLS/SSL, where the security protocol delivers application layer and operates at the transport layer. The applications must be written to support SSL/TLS. The M-SE native mode delivers transparent transport layers, so every application based on UDP or TCP can be used without any adoption. It can be compared to transport mode of IPSec. This mode can be used to secure a connection in an efficient way to dedicated server, like FTP, HTTP. The advantage of native mode is the simplicity of implementation. Because of its simplicity, the usage is very restrictive. It cannot be used with any protocols transporting IP addresses in their payloads. The M-SE does not take care of proper translation of the payloads. Bundled sessions are also not supported. M-VPN concentrates on tunnel mode and native mode is a minor feature.

The M-SE contains a procedure for updating the PoA parameter described in 3.11. The procedure is implemented with notification messages (see 5.8.1). It is build of request and reply message. The messages are encapsulated in the M-SE protocols and therefore protected.

The outbound processing of M-SE is different from the classical protocols, like IPSec. The main problem is how to match the packets to the M-SE session. The IP addresses of the Mobile Nodes cannot be used since they are dynamic. For this reason, a new concept of Mappers is introduced. A Mapper is a constant IP address assigned to an M-SE session and used for transport between the Tunnel Node and Mobile Node. The working principle is explained in the following 3.6.1. The inbound processing is similar to IPSec, where the packet is matched using unique session ID.

3.6.1 Mappers

The Mobile Node and Tunnel Node can be deployed at different physical hosts, as already described in 3.3.2. For the communication between physical hosts, IP addresses are

required. The public IP addresses of the Mobile Nodes cannot be used since they are dynamic. The assigned Home IP addresses are encapsulated in the tunnel and are unknown before the tunnel establishment (causality principle). Local IP addresses, called Mappers, are used to solve this issue. The Mappers are IP addresses used between the Mobile Node and the Tunnel Node for transporting packets. The Mappers replace the dynamic transport IP addresses. They only have local meaning between the MN and TN and are fully transparent to the rest of the hosts. In native mode, they are IP addresses between the application server and the Mobile Node. The Figure 3.10 presents the layer structure in native mode. The Figure 3.9 gives the layer in tunnel mode.

The node assigns two IP addresses for every M-SE, called session and termination Mappers. The addresses are constant during the session. The termination Mapper IP is the address of the Tunnel Node in tunnel mode or of the application server in native mode. The session Mapper IP is the IP assigned to every M-SE session. The session Mapper must be locally unique in the network routing domain between the Mobile Node and tunnel/termination server. The session Mapper assignment can be carried out depending on the ID used in M-KE or on the pre-defined pools. The session Mappers can be used as user identifiers by the TN or application server. For example: the administrators receive different Mappers to the normal users.

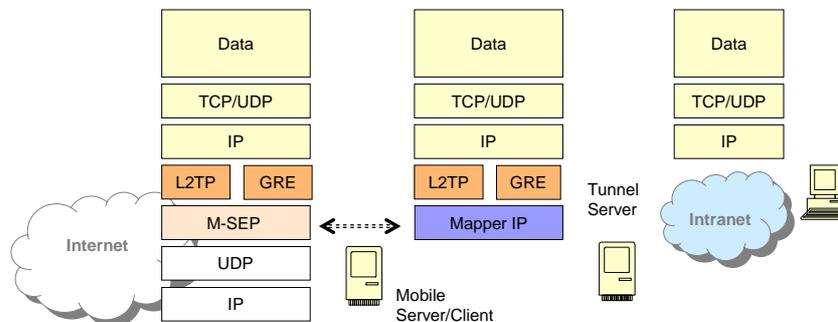


Figure 3.9: Mappers in tunnel mode

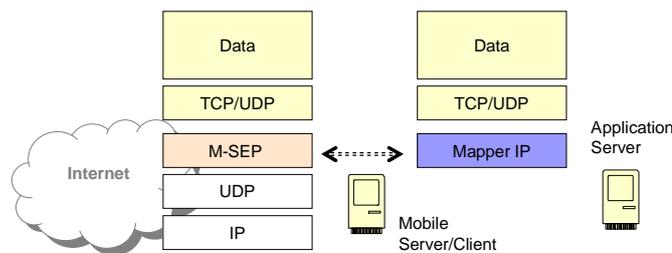


Figure 3.10: Mappers in native mode

The network of the Mappers is closed and therefore, private IP's can be used. The Mappers are not known or announced to the end hosts. They have local meaning for the configuration of the MN and TN. In general, the mapping is a way to overcome the issues due to the logical separation of tunnel and secure protocol. The separation of the tunnel and secure server has multiple advantages described in 3.3.1 and chapter 5.

3.7 Anti-tracing mechanism in M-SE

A major requirement for the secure IP mobility is protection against tracing of the Mobile Nodes (MNs) movements. Tracing of movements means physical location of the MN during the session as described in 2.6. This is achieved through mapping between header values and transport IP addresses. The IP can be translated to a physical location and in this way, the

movements of the host are physically traced. The link layer is out of scope in this work since it is used only with the network broadcast domain, see 2.6.

Every predictable and unique header value can be used for the tracing. A typical example is using the session identifier to make the match. The value is unique and constant during the session. The sequence numbers can also be used, since they are predictable and semi unique. The probability for two sessions with the same sequence values is low and trace will probably succeed.

The main task is to remove all predictable and unique fields from the header. First, the M-SE header is split in to two areas: encrypted and unencrypted. The encrypted part contains all information not directly required for the matching of the M-SE packet to certain session, like sequence number. The sequence number is not decisive for matching of the M-SE packet to session and will be encrypted. The unencrypted header contains the values necessary for inbound processing of the M-SE and other not unique values. The session identifier is the only unique value required for the inbound processing. Without the identifier, the packet cannot be mapped to the right session.

The idea is to make the session ID dynamic. The value must change during the session in unpredictable way. The changes must be made easily without negotiation or other CPU intensive operations. The introduced method is called jumping session ID. An eavesdropper can read the header values, but they are pseudo random. The header is not constant during session and cannot be matched the IP.

There are two different session IDs for the inbound and for the outbound M-SE session. The inbound session ID is defined by local host. It must be unique for the host since it is used to map to incoming packets to the right M-SE session. The outbound session ID is defined by the remote peer and used by sending. These two session IDs are announced to each other during the M-KE.

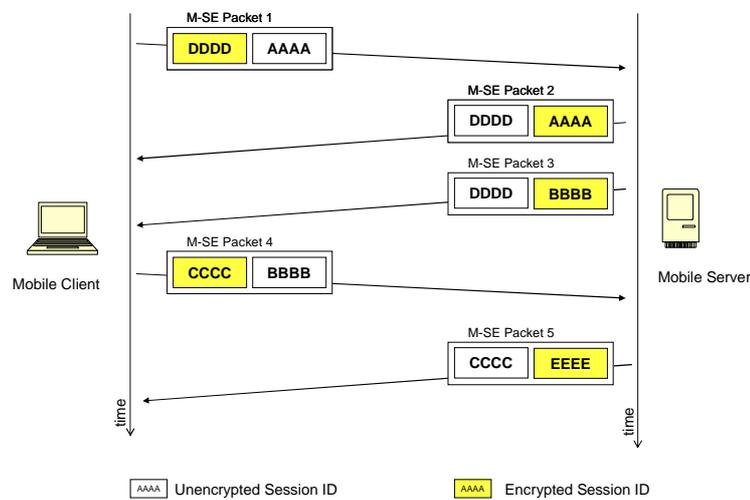


Figure 3.11: Jumping session ID

The target is to allow the local host to change the inbound session ID anytime during the M-SE connection. The outbound session ID can be changed in similar way by the remote peer. For this purpose, each M-SE packet contains two session IDs. The first one is part of the unencrypted header. It is the session ID used by the receiver for identification. The second session ID is embedded in the encrypted header, thus it is protected. The protected session ID is part of the encrypted header in M-SE between the MC and MS. The value is protected as the application data. The second session ID announces an alternative values which will be accepted by the sender (inbound). The receiver must use this session ID by sending packets and erase previous values. The announced session ID exchanges the current one. The

outbound M-SE packets are used to announce new inbound session IDs in this way. The announced session ID can be changed with pseudo random algorithm at every packet. The algorithm is left free to the implementation. The intermediate attacker taps only the unencrypted header and therefore, cannot follow the change of the session ID. The Session ID seems random for the attacker and it cannot be matched to IP.

The working principle, called jumping session ID, is shown in practical terms in Figure 3.11. The Mobile Client (MC) first sends M-SE packet 1 with the unencrypted session ID AAAA and the encrypted ID of DDDD. This means the next packets sent by Mobile Server (MS) must use session ID DDDD in the unencrypted header part. The MS uses the new session ID DDDD in packet 2. The MS do not announce new session ID, thus is keeps AAAA. The MS announces new session ID BBBB in the next packet 3. The MC uses the received session ID BBBB and announces new session ID CCCC in packet 4. The MS uses CCCC for the outbound session ID in packet 5.

Lost packets lead to missing announces of session ID. If the receiver has not obtained a message with a new session ID, then the host cannot use it. The session will be lost. The same effect can occur by out of sequence packets. The old packets containing announces can change back the session ID. Synchronization through acknowledgment is not very reasonable for security reasons. The announce and the acknowledgement message follow shortly in time. An attacker can easily follow this property and map these messages to one M-SE session. The proper session ID can be extracted from the unencrypted header. For this reason, the algorithm is enchased to tolerate losses and out-of sequence packets without synchronization.

A window of tolerance is deployed to avoid drops because of asynchronous states. The window of tolerance is simple principle saying that the host must accept packets with old inbound session IDs. If the packet with the announced session ID gets lost, the remote peer will continue using old session ID. The received packets will be matched property, since the host accepts past session IDs as well. Out of sequence packets do not lead also to failure at the M-SE. Because of physical limitations, not all past session IDs can be stored. A window of the last 64 values is defined in M-SE. Every MN uses single outbound session ID but accepts 64 inbound session IDs per M-SE connection.

Out of sequence or lost packets are not always due to network failures. They are normal behaviour in network implementing quality of Service (QoS). All backbone networks at the carrier are using QoS currently. The QoS defines priority by the packet processing. For example, VoIP packets have typically higher priority then TCP traffic. When the bandwidth is utilised, the router starts to drop the packets or to use low priority queues with a big latency.

The MS and MC can use multiple destination and source ports in parallel to session jumping techniques. They can rotate some of them on random principle as described in chapter 4.

It must be underlined that every session hiding technique has it limits. There must be a sufficient number of parallel M-SE sessions, so that identification becomes impossible. If there is a single session in a IP domain, then it can be easily identified.

3.7.1 Alternative method for session hiding

An alternative method is using generator of pseudo random numbers for automatically exchanging the session ID and sequence number. The initialisation vector for the random generators is delivered in secure way during the M-KE. The Mobile Client and Mobile Server deploy the same pseudo random algorithm with the same initialization vector. Both nodes generate the same numbers unknown to an intermediate attacker. The sequence number and the session ID can be exchanged by this single number. The new session ID represents the sequence number too, since every pseudo random number has defined sequence index by the

generation. For example, the 7th generated number is 123212. By getting the new Session ID (random number) the Mobile Nodes proves, which sequence index it has. Lost or out of sequence packets can be handled using a window of tolerance as in the previous section.

The advantage of this method is the reducing the header size by representing the session ID and the sequence number by one single field. The random number generation requires CPU resources, which must be considered. The major disadvantage of this method is that the new session ID is not unique for the local host. The inbound session ID must be unique at the local host. If the Mobile Node has multiple parallel M-SE session, then the suggested inbound session ID can overlap in certain time from mathematical point of view. There is no way to assure that two random generator do not produce the same numbers for some time. The received packet cannot be matched to the right M-SE sessions. Even if the probability for overlapping can be reduced it is present. This major disadvantage convinced the author not to deploy this method, although it seems an elegant solution.

3.8 Dynamic server discovery

The feature allowing the dynamic overlay topology in 3.3.3 is the dynamic server discovery. It allows building dynamic structures, facilitates the load sharing and enables high availability through redundancy. The feature is part of the M-KE protocol (chapter 4) and it is realised using two notifications – announce and discover.

The MC can announce that its host is also running a MS through an announce notification. The notification can be sent only from MC to MS. The notification contains the ID of the server, supported algorithms, lifetime, resources etc. The resources can be network parameters or applications, such as a certain media server. The MS adds this entry to its lists. The PoA parameters are part of the entry. The PoA is dynamic and must be updated synchronously with the M-SE session. When the M-SE session is terminated, the entry is marked as not currently active. The entries in the list have lifetimes and when these expire, the entry must be removed.

3.8.1.1 Security consideration

The server list is distributed from authenticated Mobile Server. The list contains Mobile server providing some resources. The elements in the list are not proved by the distributor and there is no guarantee for any parameter of the entry. When the client establishes a connection, it must authenticate the Mobile Server and do not consider that each element in the list is trusted.

An insider, a member of the same administrative domain, can generate a malicious list. It can sabotage the overlay network in this way. To reduce this possibility, every item of the list can have a signed tag by trusted CA authority. The tag can be verified by the client prior to establishing a connection.

3.9 Security Associations Management Database (SAM-DB)

The Security Associations Management Database (SAM-DB) is a central interconnection point between the M-KE, M-SE and M-LU. The database includes information about current sessions in M-KE and M-SE. The main structure of SAM-DB is defined here to achieve a better interoperability between the implementations.

A unique element in the SAM-DB is the Mobile Node's unique ID. The ID can be in the form of Distinguished Name (DN) [21] or any other unique Uniform Resource Identifiers (URI) [20]. In the M-VPN, it is considered that the DN is a very suitable format for MN's ID. It represents very good companies organisation and it is already integrated in the digital

certificates [22]. For example DN can be “CN=joe.smith, OU=department A, O=vodafone, C=DE”. The ID is used in the ID for identifying the AAA rules.

The SAM-DB defines the AAA policy (Authentication, Authorisation and Accounting), which consists of rules with matching patterns (regular expression). The rules match the Mobile Node’s ID and define how the session can be authenticated. The following examples demonstrate the ideas:

An example for the use of symbol “.” (any char): “CN=., OU=TNS, O=Vodafone, C=DE” matches every string having CN parameter and matching OU, O, C. For example “CN=abcse123, OU=TNS, O=Vodafone, C=DE”*

There is an assigned priority at every rule. The rules are evaluated from the highest to the lowest priority. The evaluation of the rules is stopped by the first match. A dedicated AAA group is assigned for every rule, which must be used for this session. A part of the AAA group is the cache time for the authentication and authorisation. The parameter is a requirement following from the polling negotiation described in 3.4. The value is manually configured in the entry and can be overwritten by authenticated reply.

The corporate requirements are manifested in the rules. There are usually different group of users with different authorisation and authentication policies. The ID matching can be compared with firewall rules (ACLs) on ID basis. The AAA policy is general part of SAM-DB.

The Mobile Secure Policy (M-SP) is also part of the SAM-DB. The M-SP defines which traffic and how it must be protected. It has similar function as the Policy in IPsec [5]. There is M-SP object for every Mobile Node or group of nodes. The M-SP includes the traffic selector, IP addresses of the peer, optional Mappers, etc. They are used to maintain the M-KE and M-SE negotiation with this host. The relation between the SAM-DB, M-KE and M-SE is shown in Figure 3.12. Each M-SP entry must contain sufficient information to identify and establish M-SE. There are the traffic sectors, encryption, digest algorithms, Diffie-Hellmann groups etc.

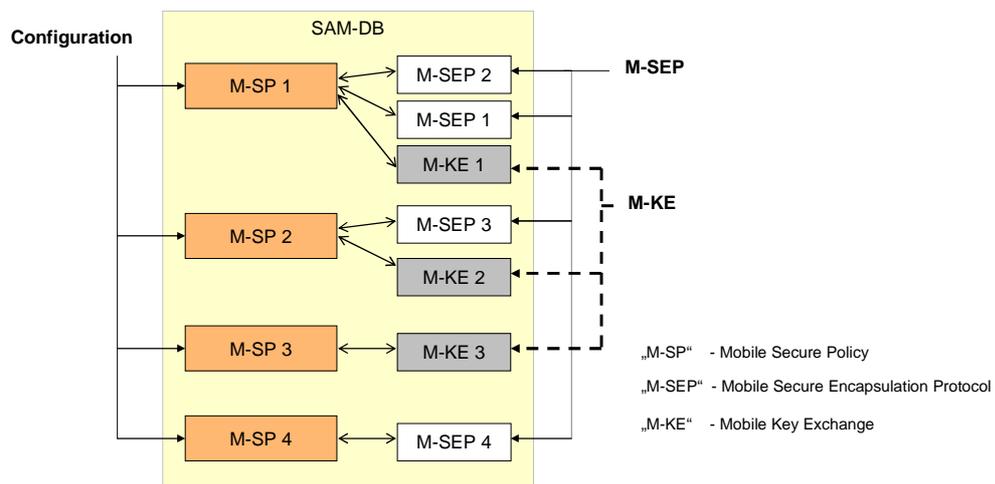


Figure 3.12: SAM-DB general structure

Every M-SP entry can be linked to multiple M-SE and to the single M-KE entry. The M-SE entry represents a set of inbound/outbound M-SE sessions to one MN. The M-SE entry is the result of M-KE negotiation. The M-KE entry contains all parameters for one negotiation, like pooling intervals, cookies, DH values etc. The M-KE is removed after the negotiation is

finished. Multiple M-SE entries to single MN is typically the result of interrupted operation described in 3.10.

Every outbound packet is matched against the traffic selectors of the M-SP entry. M-SE entry(ies) are linked to the M-SP entry. If there are multiple M-SEs, then the session with the longest lifetime is used. If there is no M-SAP entry at the MC, then M-KE negotiation is started. The result of the M-KE negotiation is M-SE entry, which is linked to the M-SP. In the example in Figure 3.12: there is not M-SE session for M-SP 3 and one M-KE 4 is started. After finishing the M-KE the M-SE entry is set and the M-KE entry removed. For example: the M-SP 4 in Figure 3.12.

By the inbound processing, the M-SE session ID and the M-KE cookies are extracted from the received packet. The values are unique for all local sessions and used to find the right M-KE/M-SE entry. Then the processing is carried out according to the M-KE or M-SE parameters.

3.10 Non interrupted operation in M-SE

Every M-SE session has a lifetime, which is a consequence of the session key straight, authentication and authorisation policy. The M-SE session must be erased after expiration of the lifetime. For the establishment of the new M-SE, the MN must successfully finish M-KE negotiation. The M-SE is not created whilst the M-KE is not finished, thus no application data can be sent and received.

To achieve uninterrupted communication during the M-KE negotiation, an overlapping of multiple M-SE sessions is allowed. The client starts new M-KE negotiations a sufficient time before the current M-SE expires. There are two concurrent M-SE sessions - one shortly before expiration and one just started. The sender must always use the one with the longest lifetime. The receiver accepts packets from all valid sessions, thus the new and legacy one. After expiration of the M-SE, the MNs are already using the new M-SE. An example of concurrent M-SE sessions is shown in Figure 3.12, M-SE 1 and M-SE 2. The parallel sessions use the same Mapper, so that the negotiation is transparent for the next (tunnel) server.

3.11 Location update notification in M-SE

Every Mobile Client is responsible for updating its PoA at the Mobile Server. The procedure for updating the PoA and getting a feedback, if the PoA has changed, is called location update. The procedure consists of request/response messages and is part of M-SE. The messages are coded in notifies described in 5.8.1.

The receiver of the request, thus Mobile Server, extracts the PoA and stores the result in the SAM-DB entry. The feedback, if the PoA has changed between the current and the last packet, is part of the response message to Mobile Client. The M-SE packets contain a flag (bit) indicating if the PoA has changed. The flag is inverted every time a PoA change has occurred (see 5.8.1).

The M-LU protocols in chapter 7 defines the frequency of the location updates, thus when to perform a location update.

3.12 Dead peer detection and NAPT keep-alive

In M-VPN, there is no need for dead peer detection or nat-keep-alives, as required in IKE for example (see A.2.2.10.2). The dynamic intervals of the location update adjust to the disconnections, regardless of the cause - by an intermediate NAPT device or by dead peer.

Independent of the disconnection reason, the location update are adjusted according M-LU described in chapter 7.

The Mobile Nodes moves from networks with and without NAPT. The M-VPN operates in the same way with or without NAPT in the network and therefore, there is no need to detect the NAPT existence. This is a big problem for IPsec (IKEv1 and IKEv2) described in chapter 2.

The M-LU covers dead-peer-detection and nat-keep-alive in efficient advantageous way. These two features are implemented separately in most of the classic protocols, like IKE. They generate significantly more load as in M-VPN (see chapters 7, 8, 9 and 10).

3.13 TCP in TCP Tunnel

The tunnel protocols implement UDP for the outer header. The use of the TCP protocol in the outer header is not suitable and leads to undesired connection blocking. The layer structure of TCP in TCP is shown in Figure 3.13. A common mistake is the use of TLS/SSL as IP tunneling protocols. If the user sends TCP data, then there are two TCP headers encapsulated in each other, see Figure 3.13.

The TCP stack has the property of recovering lost data and congestions control. In general, the packet loss is assumed to be caused by overloaded links. When there are losses, the TCP windows size is reduced, which decreases the transmission rate [27]. Overloads of the link are avoided in this way. The best example is starting concurrent TCP sessions to the same destination from the same host. Even started at the different time points, the session will have the same speed after some adoption time. The TCP's mechanism for speed reducing on losses leads to fair balance of the connections. Depending on the speed, the lost packets are retransmitted.

In the TCP in TCP, there are two stacks responsible for the re-transmitting of the same physical packet and adjusting the same physical speed. The lost packet will be retransmitted twice by the different stacks. If the stacks have different window sizes, this will lead to different speeds of retransmission. For example: the outer TCP header resends in 10 sec and the inner TCP in 2 sec. This leads to overloading of the internal buffer and blocking of the connection. There is a clear circular dependency. TCP in TCP is a very bad implementation style, because the connection can easily overload with some losses. For this reason, TLS/SSL is not proper protocol for encapsulation of IP packets including own TCP header.

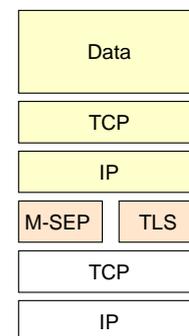


Figure 3.13: TCP in TCP

3.14 Summary and contributions

The Mobile VPN protocol has many new features improving the mobility and security. The protocol targets primary remote VPN access in mobile environment, but it can be used in other overlay networks. The highlight properties can be summaries as:

- Redundancy and load sharing capabilities with using Tunnel Node, see 3.3.2
- Dynamic overlay network topologies thought resource distribution, see 3.8.
- Anti-tracing mechanism of Mobile Node's location, see 3.7
- Key exchange during the Mobile Node changes it's PoA (Polling), see 3.5.2
- M-VPN integrates in the existing AAA structures, see 3.9.
- Dead peer detection and Nat-Keep-alive thought PoA update procedure, see 3.12

- The separation of Mobile Node and Tunnel Node enables excellent integration in the existing infrastructure, see 3.3.1

3.15 References in chapter 3

- [1] 3GPP TS 22.101: "Universal Mobile Telecommunications System (UMTS): Service aspects; Service principles"
- [2] Mogul, J., and S. Deering, S., "Path MTU Discovery", RFC 1191, DECWRL, Stanford University, November 1990
- [3] Postel, J., "Internet Control Message Protocol", RFC 792, STD 5, September 1981
- [4] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002
- [5] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005
- [6] Tzvetkov, V., Zuleger, H., "Service Provider implementation of SIP regarding security", IEEE proceeding, 2007
- [7] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowitz, "Extensible Authentication Protocol (EAP)", RFC 3748, June 2004.
- [8] Jonsson, J. and B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", RFC 3447, February 2003
- [9] Postel, J., User Datagram Protocol. RFC 768, NIC, August, 1980
- [10] Postel J., Internet Protocol. RFC 791, NIC, September, 1981.
- [11] Srisuresh et al., "Traditional IP Network Address Translator (Traditional NAT)" RFC 3022, January 2001
- [12] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Methodology for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", draft-ietf-mmusic-ice-13 (work in progress), January 2007.
- [13] Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn, G., and B. Palter, "Layer Two Tunneling Protocol L2TP", RFC 2661, August 1999
- [14] Johnson D. et al, "Mobility Support in IPv6", RFC 3775, June 2004
- [15] Perkins C., Ed., "IP Mobility Support for IPv4", RFC 3344, August 2002
- [16] Plummer, D., "Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware", STD 37, RFC 826, November 1982.
- [17] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, December 1990.
- [18] "Intermediate System to Intermediate System Intra-Domain Routeing Exchange Protocol for use in Conjunction with the Protocol for Providing the Connectionless-mode Network Service (ISO 8473)", ISO DP 10589, February 1990
- [19] Hedrick, C., "Routing Information Protocol", RFC 1058, Rutgers University, June 1988.
- [20] Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", RFC 3986, January 2005.
- [21] ITU-T Recommendation X.501: Information Technology - Open Systems Interconnection - The Directory: Models, 1993.
- [22] ITU-T Recommendation X.509 (1997 E): Information Technology - Open Systems Interconnection - The Directory: Authentication Framework, June 1997.
- [23] Rigney, C., Willens, S., Rubens, A. and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000
- [24] Neuman, C., Yu, T., Hartman, S., and K. Raeburn, "The Kerberos Network Authentication Service (V5)", RFC 4120, July 2005
- [25] Finseth, RFC 1492, "An Access Control Protocol, Sometimes Called TACACS", July 1993
- [26] Patel, B., Aboba, B., Dixon, W., Zorn, G. and S. Booth, "Securing L2TP using IPsec", RFC 3193, November 2001.
- [27] Postel, J., "Transmission Control Protocol," STD 7, RFC 793, September 1981.
- [28] Farinacci, D., Li, T., Hanks, S., Meyer, D. and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000.
- [29] Gelb, A., "Applied Optimal Estimation", MIT Press, 1974, ISBN: 0262570483
- [30] Wiener N., "The Extrapolation, Interpolation and Smoothing of Stationary Time Series," John Wiley & Sons, Inc., New York, N.Y., 1949.
- [31] Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp T., "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking", IEEE, VOL. 50, NO. 2, 2002
- [32] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

4 Mobile Key Exchange

The Mobile Key Exchange (M-KE) protocol negotiates cryptographic parameters, keys and authenticates the participants as described in chapter 3. The protocol runs over UDP using port 733 at the Mobile Server (MS). The port number must not be used by any other application. The motivation for port 733 is because there is no registered application for this port in IANA [15] currently. Different port could also be used. The M-KE negotiation is initiated by the Mobile Client according to the Mobile Secure Policy (M-SP) explained in 3.9. The values of the M-SP required to start M-KE negotiation are:

- Fully Qualified Domain Name (FQDN) of the Mobile Server. The Mobile Client resolves the FQDN to A record in DNS. If there are multiple IPs, the client must try sequentially to establish connection to all of them until obtaining a response of the Mobile Server. There is negotiation timeout defining the time before switching to the next IP. The direct use of IP addresses instead of FQDN is possible, but undesired since it leads to inflexibility.
- The ID of the local host and remote peer must be defined. The ID must be Distinguished Name [16] or URI [17]. The local ID must be exact defined. The remote ID may be defined using pattern, thus regular expressions.
- A list of supported cryptographic algorithms must be set. For example: CBC AES 128 Bit with SHA integrity protection with 3600 sec live time.
- The authentication rules based on ID must be set as defined in 3.9. Ability to authenticate the remote peer must be present.

The output of the M-KE negotiation is the parameters for the M-SE session establishment. They are stored in the M-SAB database in M-SE entry. The mandatory values are:

- Algorithms for encryption and integrity protection of the M-SE session must be set.
- Four keys for encryption and authentication of M-SE sessions (inbound and outbound)
- Peers IDs
- Inbound and outbound Session IDs must be set according 3.7.
- M-SE session life time
- The shortest interval between the movements supported by the MNs
- Resources at the MS
- Optionally, a list of other MSs according 3.8
- Transport parameters, like PoA of the peer, defined in 3.9.

4.1 Authentication methods

M-KE is very flexible regarding the authentication. The MC and MS agree on any combination of the following authentication methods:

- Signature authentication with RSA [5] and DSS [6] using x509 certificates
- Password authentication similar to CHAP [7] authentication in PPP

- Shared secret authentication
- Extensible Authentication Protocol (EAP) [8]. The EAP is encapsulation for authentication protocols and optional master session key derivation. Using EAP in M-KE is very suitable when the Mobile Node must be integrated in infrastructure supporting specific EAP authentication. For Example: SIM authentication can be performed using EAP-SIM [9] and in this way some interworking between M-KE and the existing GSM systems. The EAP must be considered only when the M-KE integrated methods cannot be used, thus signature, shared secret and password (The IKEv2 [13] also provides build-in methods and EAP). To avoid circular dependencies, since the EAP is only encapsulation, the following combination can be used in M-KE (see discussion in 6.1):
 - When the MC and MS use EAP for mutual authentication, than only EAP methods with Master Session Key (MSK) derivation can be used. These are: EAP-TLS [19], EAP-PSK [20], EAP-FAST [21], EAP-SIM [9] and EAP-AKA [18].
 - When only one of the Mobile Nodes (MC or MS) is using EAP and the other the integrated M-KE authentications, there are no restrictions for the EAP method. The integrity of the negotiation is verified by the Mobile Node authenticated by EAP, since it receives authenticated HMAC. The case is similar to TLS [1] with server side certificates.
 - When both Mobile Nodes are using the M-KE integrated authentication methods, there are no restrictions.

4.2 M-KE messages

The M-KE consists of four message types: *ClientHello*, *ClientResponse*, *ServerHello* and *ServerResponse*. They are indicated in the *TypeOfMessage* byte of the M-KE header. The packets have the same structure, but they contain different payloads depending on their purpose. The M-KE is designed to achieve negotiation with a minimum number of packets. M-KE exchange without any retransmission and pending is shown in Figure 4.1.

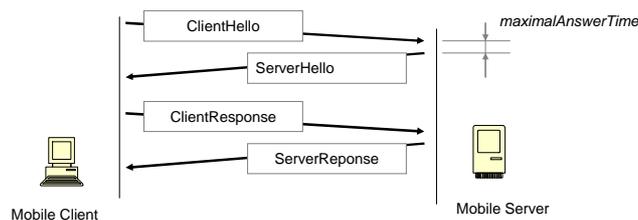


Figure 4.1: Mobile Key Exchange

The first two packets (*ClientHello* and *ServerHello*) set up an encrypted channel for further communication. The following encrypted messages (*ClientResponse* und *ServerResponse*) authenticate the peers, makes message integrity check and set up session parameters. Brief description of the packet is following:

- *ClientHello* suggests algorithms for authentication, integrity protection and encryption. The authentication algorithms may different be for the client und server. The message contains the clients DH values.

- *ServerHello* is the response of the Mobile Server to the Mobile Client. The message accepts certain encryption, hash and authentication algorithms. The Mobile Server also sends its DH values. It may request certificate from a certain CA, sent EAP values etc. Parts of the ServerHello are encrypted with the result of the DH key exchange.
- *ClientResponse* gives M-SE parameters, like session ID, live time etc. This is the packet containing Client ID. It may contain also further authentication payloads. The message payloads are encrypted. The message contains also authenticated integrity check of all exchanged packets (HMAC).
- *ServerResponse* is contains the Mobile Server parameters for the M-SE connection, Server ID, and authenticated integrity check of the negotiation (HMAC).

If the Mobile Server cannot reply within the maximum answer time defined by the Mobile Client, it replies with status pending. The client resends the message in the interval given in the pending status response, as described in 3.5. Single M-KE negotiations can consist of multiple retransmissions as the example in Figure 4.2 shows.

The participants can send notification payloads at every stage of the negotiation see 4.13.17. The HMAC payload must be included in the M-KE when notifying. The notification must be part of the encrypted payloads if encryption channel is set. Otherwise, it may be included in the unencrypted payloads. This is typically the case when the authentication fails, or there is no proper protection algorithm. To facilitate the message description in the following sections, the possible notification payload is not discussed.

The M-KE negotiation must be finished within a certain maximum negotiation time. Otherwise, the M-KE session is removed.

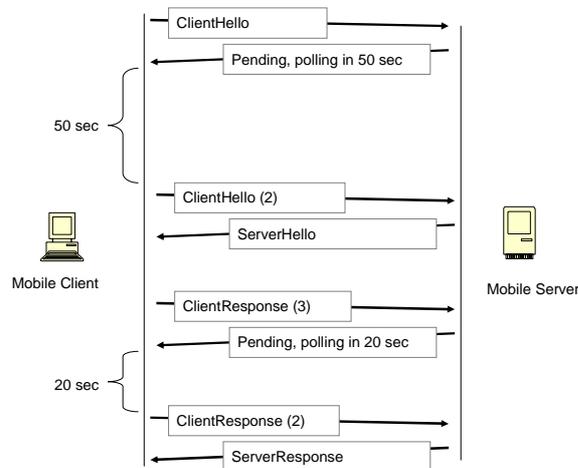


Figure 4.2: Example of M-KE negotiation with polling

4.3 Packet structure

The protocol structures are described with the notation of SSL and TLS defined in [1]. The general packet structure is:

```

struct {
    MkeHeader          header;
    Payload            mobileKePayloads[0..2^16-1]; /* unencrypted payloads */
    Payload            encryptedPayloads[0..2^16-1]; /* encrypted payloads */
} MobileKE

struct {
    ProtocolVersion    mobileKeVersion;
    TypeOfMessage      type;
    Length             len;
    opaque             cookieServer[4];
    opaque             cookieClient[4];
} MkeHeader

struc {
    uint8              version;          /* major | minor */
} ProtocolVersion

enum {
    clientHello(1), ClientResponse(3), ServerResponse(4), serverHello(5), (255)
} TypeOfMessage

struc {
    uint8              fragment;
    uint16             packetLength;
} Length

struc {
    uint8              type;
    uint8              subType;
    opaque             data[0..2^16-1];
} Payload

```

Every M-KE packer has this structure. Structures and values may be empty in some packets when there is not data for them. For example: in the *ClientHello*, there are no *encryptedPayloads* since there is not key material.

The M-KE header begins with the protocol version (*ProtocolVersion*) of one byte. The first 4 Most Significant Bits (MSB) indicate the major version. The major version is one in this specification. The 4 Last Significant Bits (LSB) indicate the minor version and are set to zero in this specification. In order to communicate, two hosts must use the same major version. The highest major version for the M-KE protocol is 7. The values from 8 to 16 are reserved for the M-SE protocol as described in 3.4.

The next byte value *TypeOfMessage* gives the general description of this message. There are four types of messages described 4.2. The following *Length* structure (see 4.7) contains the length of the M-KE message and optional fragment details. The first two bytes (*packetLength*) give the total packet length including all fragments. The maximum size is $2^{16}-1$ bytes. The *fragment* byte gives the fragment sequence number.

The next two values, *cookieServer* and *cookieClient*, contain 4 byte cookies. The client sets the *cookieClient* and the server sets the *cookieServer*. The values identify the messages. They are used to pick up the right M-KE negotiation. The cookies are the same for one negotiation and must be unique for the host. The unique values assure replay protection.

Furthermore, pseudo random cookies limit simple Denial of Service attacks. Pseudo random and unique values are recommended.

A payload is a data structure containing related information, like encryption algorithms. The payloads carry all data needed for the negotiation. The sequence of the payloads in the messages is free for the implementation. The payload has a general structure consisting of *Type*, *SubType* and *Data* field. The *Type* byte specifies the information carried in the payload, like certificate. The *Subtype* gives further information, such as the encoding type of the data. It can also include value itself, like the Diffie-Hellman group number. The *Data* vector contains the data, like the certificate. The data field is optional, since only the existence of the payload is decisive in some cases.

The payloads are grouped in two vectors unencrypted and encrypted. Security relevant payloads are embedded in encrypted vectors, such as the ID of the user. The premise for using the encrypted vector is the existence of the shared master session key (secret). The unencrypted vector carries payloads for key derivation, like Diffie-Hellman public values and protection algorithms. The payloads are exactly defined in chapter 4.8.

4.3.1 ClientHello

The first M-KE packet sent by the Mobile Client is the *ClientHello* packet. It has the following structure:

```

struct {
    MkeHeader          header;
    struct {
        DhValuePayload      clientDhValues[3..2^16-1];
        ProtectionAlgPayload clientSuggestedProtectionAlg[3..2^16-1];
        ProtectionAlgPayload serverRequiredProtectionAlg[3..2^16-1];
        MaxAnswerTimePayload requiredMaxAnswerTime;
    } UnencryptedPayloads;

    struct {
        /* no encrypted payloads */
    } EncryptedPayloads;

} ClientHello;

struct {
    ProtocolVerion      mobileKeVersion;
    TypeOfMessage       clientHallo;
    Length              len;
    opaque              cookieServer[4]=0x00000000;
    opaque              cookieClient[4];

} MkeHeader;

```

The header contains *TypeOfMessage* set to *clientHallo*. The Mobile Client generates the *cookieClient* value of 4 bytes. The *cookieServe* is null. The *Lengths* is set according to 4.3. The *UnencryptedPayloads* contains multiple payloads defined in 4.8. The *clientDhValues* contains the list of *DhValuePayload* payloads see 4.13.1. The client sends one payload for each DH group supported, including public values. To reduce the number of exchanged messages, DH groups are predefined with large primes and elliptic curve equations. The node must only send the group number and its public value. For example: If the client proposes four groups, then the four payloads with four public values are sent. This generates more resources, since only one group with its public values is chosen. The benefit is in reducing the number of negotiation packets, which is considered critical in a mobile environment. The

server chooses one group by responding with a single *DhValuePayload* containing its public value. The master session key (secret) generation of DH values is defined in 4.6.

The next vector *clientSuggestedProtectionAlg*, see 4.13.2, contains the protection algorithms offered by the client, i.e. encryption, integrity check and authentication. For example, the client offers authentication with a password and can send packets encrypted in CBC_AES 128 bit, integrity protection with HMAC with SHA1. The client and server agree on supported algorithms using these Payloads. There is not session key at this point and the payloads are sent in clear. The Client ID and Server ID are not sent in the first two packets since they must be protected. The single structure *ProtectionAlgPayload* contains one combination for authentication, encryption and integrity protection. More combinations are set in multiple payloads.

The vector *serverRequiredProtectionAlg*, see 4.13.2, contains the required protection algorithms from the server. This is a list of expected algorithms, for example the client expects RSA signature authentication for the server authentication and CBC_AES 128bit, HMAC with SHA1. The separation between offered and accepted algorithms gives maximum flexibility, like server authenticates with certificate and client with password. The server chooses one combination and replies with single payload in *clientSuggestedProtectionAlg* and *serverRequiredProtectionAlg*.

The *requiredMaxAnswerTime*, see 4.13.3, gives the maximum packet processing time at the server, tolerated by the mobile client (see Figure 4.1). The client can omit this payload if it has not estimation of this value. Knowing an estimated value of the constant PoA of the client reduces the polling interval. For example: if the PoA change is expected to be once a day then the server can certainly take 30 sec to generate the reply.

The *EncryptedPayloads* structure is empty since there is not key material. The structure is listed only for completeness.

4.3.2 ServerHello

After the server receives the *ClientHello* packet, it creates a new M-KE negotiation entry. It generates its *cookieServer* value and uses it in all packets for this negotiation. The answer structure is the following:

```

struct {
    MkeHeader          header;

    select (ReplyStatus) {

        case pending:          /* no immediate reply possible */
            struc {
                PendingPayload pending;
            } UnencryptedPayloads;

            struc {
                /* no encrypted payloads */
            } EncryptedPayloads;

        case replyReady:      /* reply ready */

            struc {
                DhValuePayload          serverDhValue;
                ProtectionAlgPayload    clientProtectionAlg;
                ProtectionAlgPayload    serverProtectionAlg;
            } UnencryptedPayloads;

            struc {
                MinPoaChangeIntPayload  minPoaChangeInterval;
    
```

```

    MaxUpdateIntPayload    maxUpateInterval;

    select (ClientAuthenticationType){
    case rsasig:
        CertificateRequestPaylaad    clientCertifctcRequest[3..2^16-1];
    case dsssig:
        CertificateRequestPayload    clientCertifctcRequest[3..2^16-1];
    case password
        :
        ChallengePayload              clientChallengeValue;
    case eap
        EapPayload                    clientEap;
    case sharedsecret:
        /* no payload */
    }

    PaddingPayload          padding;

} EncryptedPayloads;
} ServerHello;

struct {
    ProtocolVerion          mobileKeVersion;
    TypeOfMessage           serverHallo;
    Length                  len;
    opaque                  cookieServer[4];
    opaque                  cookieClient[4];
} MkeHeader

enum {pending, replyReady} ReplyStatus;

enum {rsasig, dsssig, password, sharedSecret, eap ,null .. 255} ClientAuthenticationType;

```

If the server cannot reply to the client within the *maximalAnswerTime*, it sends a pending payload in *ServerHello*. In the data field of the *pending* payload is given the interval for the next polling, see 4.13.4. The value subtype can be seconds or milliseconds.

If the server can reply (*replyReady*), it selects the desired DH group and sends the *serverDhValue* Payload. The *clientProtectionAlg* value contains the selected authentication, encryption and integrity protection for the client. The client has sent a list of possibilities and the server selects only one. The server sends *serverProtectionAlg* in the same principle. These payloads are unencrypted in the *ServerHello* message. The DH values are exchanged with these payloads and the protection algorithms set, so there is sufficient information for generation of the Master Session Key (MSK), see 4.6. It must be stresses that the Master Session Key is also considered not trusted until the DH values are not authenticated. The DH values are authenticated with exchange of the HMAC values in *ClientResponse* and *ServerResponse*.

The next payloads are encrypted, as described in 4.9. The *minPoaChangeInterval* value gives the minimum interval for PoA change supported by the server. The value depends on the server performance. If the client expects to move more frequently than the server supports, then it can terminate the connection. The *maxUpateInterval* gives the idle timeout interval before the M-SE connection is deleted. If the client does not send any authenticated M-SE packet within this interval then the session is erased. The idle timeout must be announced to the parity, i.e. the client. This is very important since one of the main tasks in the M-VPN is to achieve optimum updates as described in 7.

If the client authenticates with a signature, then the server attaches *clientCertifctcRequest* vector. It contains multiple *CertificateRequestPayloads*, one for every accepted certificate authority see 4.13.8. If the client wishes to authenticate with a password then the challenge is

sent in the *clientChallengeValue*, see chapter 4.13.10. If the EAP is to be used, then the server uses *clientEAP*, see chapter 4.12.

The padding values in the last payload add empty values to the *EncryptedPayloads*, so they can be encrypted with block cipher, see chapter 4.9.

4.4 ClientResponse

After receiving the *ServerHello*, the client sends all further payloads encrypted. The structure of the *ClientResponse* is the following:

```

struct {
    MkeHeader          header;

    struc {
        /* no payloads */
    } UnencryptedPayloads;

    struc {
        IdPayload          clientId;
        MinPoaChangeIntPayload  minPoaChangeInterval;
        MaxUpdateIntPayload  maxUpateInterval;
        SessionIdPayload    clientSessionId;
        SessionLifeTimePayload  clientSessionLifeTime[0..2^16-];
        TrafficSelectorPayload  clientTrafficSelector;

        select (ServerAuthenticationType){
            case rsasig:
                CertificateRequestPaylaad  serverCertifcticeRequest[3..2^16-1];
            case dsssig:
                CertificateRequestPayload  serverCertifcticeRequest[3..2^16-1];
            case password:
                ChallengePayload            serverChallengeValue;
            case eap:
                EapPayload                    serverEap;
            case sharedsecret:
                /* no payload */
        }

        select (ClientAuthenticationType){
            case rsasig:
                CertificatePaylaad          clientCertifctice;
            case dsssig:
                CertificatePayload          clientCertifctice;
            case password :
                ChallengePayload            clientChallengeResponse;
            case eap
                EapPayload                    clientEap;
            case sharedsecret:
                /* no payload */
        }

        /* optional notification */
        NotificationPayload                notification;

        HmacPayload                        clientHmacValue;
        PaddingPayload                      padding;

    } EncryptedPayloads;
} ClientResponse;

```

```

struct {
    ProtocolVerion      mobileKeVersion;
    TypeOfMessage      clientResponse;
    Length              len;
    opaque              cookieServer[4];
    opaque              cookieClient[4];
} MkeHeader

enum {rsasig, dsssig, password, sharedSecret, eap, nul ..255} ServerAuthenticationType;

```

The *clientId* contains the ID of the client, which is a unique string. If the server has other M-KE negotiations with the same ID then the M-SE are removed after successful authentication. The *maxPoaChangeInt* is the idle timeout after which the session is deleted. The *minPoaChangeInterval* contains the minimal interval between the PoA changes supported by the client.

The *clientSessionId* contains the session ID in *SessionIdPayload*, see 4.13.14. It is used in the M-SE for session identification. The optional *clientSessionLifeTime* payloads give the lifetime of the M-SE session. The lifetime can be in seconds and/or in bytes, see 4.13.7.

The *clientTrafficSelector* contains the transport layer parameter for the protected traffic. They are used together with the Mapper for classifying the data for encryption.

Depending on the selected server authentication, the client sends *serverCertifictRequest*, *serverChallengeValue* or *serverEAP*. The *clientCertifict*, see 4.13.9, is sent when the client authenticates with a signature. The certificate must be issued from one of the authorities listed in the *clientCertificateRequest* part of *HelloServer* message. The client sends the *clientChallengeResponse* when using password for authentication, see 4.13.10. The result can be forwarded to the AAA server (Radius) for standard CHAP authentication. For EAP authentication the *clientEAP* values is used.

NotificationPayload (see 4.13.17) can be optionally embedded if the client makes server discovery or announces server service. The notification procedure is described in 4.10.

The purpose of *clientHmacValues* payload is integrity protection of the exchanged payloads. It contains the hash of the significant sent and received messages. The hash is protected by server signature or authentication key, see 4.13.16. Significant messages are all M-KE without the retransmissions and pending notifications. Only after successfully verification of the HMAC, the client can be considered authenticated and the M-SE session is created. If the verification fails, then the M-KE negotiation is considered as bogus. Optional notification can be sent to the client. How the *clientHmachValues* is built is described in *HmacPayload* in 4.13.16.

4.5 ServerResponse

The *ServerResponse* is the last message with the following structure:

```

struct {
    MkeHeader          header;

    struct {
        /* no payloads */
    } UnencryptedPayloads;

    struct {
        IdPayload      serverId;
        SessionIdPayload serverSessionId;
        SessionLifetimePayload serverSessionLifeTime[1..2^16-1];
        TrafficSelectorPayload serverTrafficSelector;
    }

```

```

select (ServerAuthenticationType){
  case rsasig:
    CertificatePayload serverCertificate;
  case dsssig:
    CertificatePayload serverCertificate;
  case password :
    ChallengePayload serverChallengeResponse;
  case eap
    EapPayload serverEap;
  case sharedsecret:
    /* no payload */
}
/* optional notification */
NotificationPayload notify
HmacPayload serverHmacValue;
PaddingPayload padding;

} EncryptedPayloads;
} ServerResponse;

struct {
  ProtocolVersion mobileKeVersion;
  TypeOfMessage serverResponse;
  Length len;
  opaque cookieServer[4];
  opaque cookieClient[4];
} MkeHeader

```

The server sends its ID in the *serverId* value. The *serverSessionId* is the M-SE session ID used by the server for sending. The M-SE session is created after successful authentication and verification of *serverHmacValue*. Otherwise, the negotiation is discarded and notification may be sent. The server sets the session lifetime in the *serverSessionLifeTime*. The lifetime must be set and at least one *SessionLifeTime* payload must be sent. The lifetime cannot be larger than the one suggested by the client. The client adjusts its value to the server ones.

The transport parameters are sent in the *serverTrafficSelector* value. The client must adjust its traffic selector to the server one if the values are different.

The authentication value *serverCertificate* contains the certificate of the server, if signature authentication is required by the server. If password authentication was used by the user then the *serverChallengeValue* is sent. The *serverEap* gives the EAP values.

The *serverHmacValue* contains the integrity check of all significant messages, which are sent and received by the server, see 4.13.16. The client must verify the value carefully since it gives the proof of private key or shared secret ownership by the server.

4.6 Key derivation

There are 6 keys derived from Diffie-Hellman values exchanged in the M-KE:

- *SKe_c* is the client encryption key of the M-KE encrypted payload.
- *SKe_s* is the server encryption key of the M-KE encrypted payload.
- *Sa_c* is the client authentication key for integrity protection of the M-SE packets.
- *Sa_s* is the server authentication key for integrity protection of the M-SE packets.

- Se_c is the client encryption key of the M-SE packets.
- Se_s is the server encryption key of the M-SE packets.

All keys must be of a fixed size required by the protection algorithm. If the protection algorithm requires specific key, such as DES and 3DES with parity bits, then this derivation from the random values must be defined in the encryption algorithm. For integrity algorithms based on hash, the key size is always equal to the HASH length.

The key derivation is the same as IKEv2 [13] and described as:

$$SKEYSEED = prf(cookieClient | cookieServer, g_{DH})$$

$$\{ SKe_c | SKe_s | Sa_c | Sa_s | Se_c | Se_s \} = prf+(SKEYSEED, cookieClient | cookieServer | clientSessionId | serverSessionId),$$

where $prf()$ is pseudo random function described in 4.6.1 (The brackets “()” denote function). The $pfs+()$ denoted pseudo random stream with sufficient length see chapter 4.6.2. The g_{DH} denotes the shared secret of the DH negotiation. The $g_{DH} = (g^s \bmod p)^c \bmod p = (g^c \bmod p)^s \bmod p$, where: $g^c \bmod p$ is the public values generated by the client, $g^s \bmod p$ is generated by the server, p and g are group predefined values see [13]. The s and c are called private values. If necessary, padding with zeros is added to g_{DH} to reach sufficient length. The $cookieClient$ and $cookieServer$ are the cookies from the M-KE header. The $clientSessionId(2)$ and $serverSessionId(2)$ are the session ID’s negotiated in the M-KE.

4.6.1 Pseudo Random Function

The pseudo random function $prf()$ is described in [12] and out of scope of this document. The function uses the same integrity hash algorithm negotiated in the M-KE for M-SE protection, thus SHA1 or MD5.

4.6.2 Pseudo Random Stream

The pseudo random stream function returns variable length pseudo random data. The keys have different sizes and the result of $pfs+()$ has the total size of all keys. M-KE uses the same $pfs+()$ as IKEv2 [13] defined as follows:

$$prf+(K,S) = T1 | T2 | T3 | T4 | \dots$$

where:

$$T1 = prf(K, S | 0x01)$$

$$T2 = prf(K, T1 | S | 0x02)$$

$$T3 = prf(K, T2 | S | 0x03)$$

$$T4 = prf(K, T3 | S | 0x04)$$

The equation can be continued, when necessary to compute all required keys. The keys are taken from the output without considering the boundaries of the $T()$ functions.

4.7 Fragmentation of Mobile KE datagram

A packet must be fragmented if it is larger than the maximum MTU of the path [2]. The UDP packet must be fragmented at IP layer as defined in [2]. This leads to problems with NAPT router described in A.2.1.9 and 3.2. To handle this issue, the fragmentation is made at M-KE layer.

If a M-KE packet must be fragmented, the data block of encrypted and unencrypted *Payload* vectors are split into fragments. The M-KE packets contain an UDP and M-KE

header and part of the payloads. The packet can be handled by the intermediate NAPT router, since the UDP header is present. The *fragment* byte in every M-KE message is set to the sequence number of the fragment. The sequence starts from 1 and increases monotone with 1 for every fragment within one M-KE negotiation, i.e. *ClientHello*, *ServerHello* etc. The numbering does not start from the very beginning on every fragmentation, since it must be unique. Upon reaching the 255, the counting starts from the beginning. The *packetLength* contains the size of the total datagram equal to the sum of all fragments. If the byte *fragment* is zero, then there is no fragmentation.

The receiver identifies the packet using the *cookieServer/cookieClient*, *packetLength* and the *fragment* values. All fragments have the same cookies, thus it is the equivalent of the ID value by IP fragmentation. The fragments belonging to one datagram are gathered until the *packetLength* is reached. The sequence number indicates how these must be joined. This is a simple and efficient method for fragmentation.

The MTU size of the interface and the PMTU mechanism gives the maximum size of M-KE packets. The nodes are moving frequently in a mobile environment. The PoA must be constant for at about three roundtrips in order to be able to perform PMTU and sent packet. The PMTU requires one round trip with a static PoA. The message exchange requires constant PoA for one roundtrip plus the processing time see 3.5.1. The processing time is defined as maximum one roundtrip, thus the message exchange requires two roundtrips. Constant PoA for ca. three round trips is not always possible. It was assumed in 3.5.2 that the PoA is constant for at least two roundtrips. When PMTU is impossible, the nodes can set the maximal MTU to 1450. The value is considered as a sufficiently small. Setting the MTU to a constant value of 1450 contains the risk of losing packets in the links with lower MTU. A constant MTU must be used if no other alternatives are possible.

4.8 Payloads

The different payloads have the same general structure as shown in the 4.3. The values depending on the context are defined in the appendix 4.13.

4.9 Encrypted payloads

The algorithm negotiated in the unencrypted payload is used for protection of *EncryptedPayload*. The same algorithm is used for the encryption of the M-SE and M-KE. The block encryption algorithms require the input with length of modulo, like 64 bit. Padding with zeroes is added in *PaddingPayload* until the required length is reached.

Keys for inbound and outbound traffic are different. The client key, *SKe_c*, and the server key, *SKe_s*, derivation is described in 4.6. The encryption algorithms are defined in 4.13.

4.10 Notification

The notification is used for informing or requesting information. The notification consists of requests and responses. There is a response to every request. The *NotificationPayload* should always be carried in M-KE message together with a HMAC payload in *EncryptedPayload*. If there is no authenticated DH key, then the notification is sent in the unencrypted part as a single payload of M-KE. If carried in the unencrypted part, the notification may be discarded by the receiver. The notification payload can be added to every packet or sent as a single payload of the notification message.

The structure of notification message in the *NotificationPayload*, see 4.13.17 is:

```

enum { serverDiscovery(1),, ...255} notificationType; /* see 4.13.19 */
struct {
    opaque          notificationCookie[4]
    uint8          notificationType;
    select (notificationType) {
        serverDiscovery:
            Server  serverDiscovery[0..255];
        serverAnnounce:
            Server  serverInfo;
        redirection:
            Server  serverParams[1..255];
        else: /* any other type */
            opaque  text[1..255];
    }
} NotificationMessage;

```

The *notificationCookie* is a unique cookie of 4 byte for the identification of a response message. The initiator generates and sets the value. The responder uses the same values in the reply. The *notificationType* gives the subject and it is defined in 4.13.19.

The following value depends on the *notificationType*. A list of servers is used by *serverDiscovery* type. The structure *Server* is defined in 4.11. A single server (*serverInfo*) is included by announcing with *serverAnnounce*. If the type is a *redirection* then a list of servers is used *serverParams*. A string can be included providing the verbal notification text. The string is variable in length with a maximum 255 characters. The text has information and debugging purposes only. The action is taken depending on the notification type.

4.10.1 ServerDiscovery notification

A server discovery is a notification payload included in *ClientResponse* and *ServerResponse* packet. The client requires a list of known servers. The server responds with a list of servers. The structure is described in 4.11. For example: if two VoIP participants are behind the same NAPT router and have client and server functionality, they can communicate directly without using a public Internet server.

The request message contains null *Server* structures. The server answers with a list of servers built in the same way as described in chapter 4.11. The server also sets itself in first position on the list. The *resource* string plays a significant role. The string describes the resource provided by the server. This can be a URI [17] or any other Text. For example: news:comp.infosystems.www.servers.unix, file://Wikipedia_Articles\URL.xml, “News etc. The client evaluates the entries and can establish connection with some of them. The exact format of the *resources* strings depends on the deployment. It is left to following documents to describe exact scenarios. The priority byte is ignored.

It negotiates a native mode and sends the encrypted *ServerDiscovery* payload in the *ClientResponse* message. The message consists of the HMAC and *ServerDiscovery* payload. The server replies with the server list in its *ServerResponse*. After this, the M-KE session is terminated.

4.11 Connection redirection

The connection *redirection* notification gives alternative servers for this session. The new servers must have the same administrative rights and offer the same resources as the current one. It can be the same physical machine reachable under a new address. This type of notification can be only sent encrypted between mutual authenticated peers.

The client must try to establish M-KE and consequent M-SE session to new server(s). The entries in the list have priority and the client must start with the highest one. If two or more servers have the same priority, then parallel M-SE connection can be established. Otherwise only one session is expected.

A possible reason for the redirection can be load sharing. For example, if the server has reached its maximum capacity then new connection can be redirected. The new server is an alternative for reaching the same resources. Load distribution on session basis can be achieved easily in this way. Optionally, the client may reconfigure to use the new server instead the old one for all future connections. The replacement of the current server in the configuration depends on the local security policy.

The *Server* structure is the following:

```
enum { IPv6(1), IPv4(2), fqdn...255} Host;

struct {
    opaque      name[1..32];
    uint8       priority;
    uint16      timeToLive;
    uint8       ipVersion;
    select (Host) {
        IPv4:
            uint8   ipv4[4];
        IPv6:
            uint8   ipv6[16];
        fqdn:
            opaque  fqdn[1..255];
    }
    uint16      port;
    opaque      resources[0..255];
} Server;
```

The first string is the name of the server. Its size is limited to 32 bytes. The next one byte gives the priority. The lower the byte values, the higher the priority. The *timeToLive* contains the validity of this entry in seconds. After the time expires, the entry must be discarded. Zero value is set to ignore this field. The *Host* value gives the type connection. It can be FQDN(Fully Qualified Domain Name), for example “server1.foo.de”. The FQDN must be resolved in the DNS with its A-Record. The information can also be an IPv4 or IPv6 address. The *port* value contains the destination port for the connection establishment. The last string resources describe the type of resources contained within the server (see example in 4.10.1)

If one server is reachable at multiple addresses, then the server is listed multiple times in the list with different priorities. If the client already has a list of servers then only the entries with the highest *timeToLive* are kept. The number of entries kept by the client is left to the implementation. It depends on the memory of the node. It should be noted that some of the entries might not be reachable because of the NATP see 3.2.1.

4.12 EAP Authentication properties

The M-KE negotiation has four messages, *ClientHello*, *ClientResponse*, *ServerHello*, *ServerResponse*. The EAP payload can be added to *ServerHello*, *ClientResponse* and *ServerResponse*. The EAP is not limited to 3 messages. If more EAP messages are required then the *ServerResponse* and *ClientResponse* are repeated until EAP is finished. These additional messages contain only *EapPayload* and *HmacPayload* in the encrypted part.

4.13 Appendix M-KE

4.13.1 DhValuePayload

The payload carries the Diffie-Hellman (DH) public parameters.

```

struct {
    uint8      constDhValuePayload;          /* defined in 4.13.19 */
    uint8      contDhGroup;
    opaque     dhPublicValue[0..2^16-1];
} DhValuePayload

```

The *constDhValuePayload* is byte value indicating the type of the payload see 4.13.19. The *contDhGroup* is a byte from the predefined DH group from 1 to 5 (*contDhGroup1* to *contDhGroup5*). The groups are defined in Oakley protocol [10] and additional in [11]. The *dhPublicValue* is the public value with a variable length, in accordance with this group.

4.13.2 ProtectionAlgPayload

```

enum { client(1), server(2) } Node;

struct {
    uint8      constProtectionAlgPayload;    /* defined in 4.13.19*/
    uint8      node                         /* client or server */
    opaque     constProtectionAlgorithms<0..2^16-1>;
} ProtectionAlgPayload

```

The protection algorithm payload contains one combination of authentication, encryption and integrity protection algorithms. It has type *constProtectionAlgPayload*, see 4.13.19. One combination is stored in vector of two bytes *constProtectionAlgorithms*. The constant values defined in 4.13. The subtype *node* can be client or server, depending who should use this method. The payload can be included multiple times in one message.

4.13.3 MaxAnswerTimePayload

```

enum { sec(1), millisec(2) } Unit;

struct {
    uint8      constMaxAnswerTimePayload;    /* defined in 4.13.19*/
    uint8      Unit
    opaque     interval<0..2^16-1>;
} MaxAnswerTimePayload

```

The payload gives the maximal time for processing expected by the Mobile Server. This payload is sent by the Mobile Client. This parameter is important for defining the mobility of the client. The *constMaxAnswerTimePayload* is defined the 4.13. The subtype *Unit* gives the unit, seconds or milliseconds of the interval.

4.13.4 PendingPayload

If the server cannot answer within the interval given in *MaxAnswerTimePayload*, then it sends a pending payload.

```

enum { sec(1), millisec(2) } Unit;

struct {
    uint8      constPendingPayload;         /* defined in 4.13.19*/
    uint8      Unit

```

```

    opaque          interval<0..2^16-1>;
} PendingPayload

```

The interval vector gives the interval for polling. The client repeats the request after expiration of this interval.

4.13.5 MinPoaChangeIntPayload

The *MinPoaChangeIntPayload* gives the minimal time interval between the PoA changes supported by the Mobile Node. In this way, the client and server avoid connection drops due to low performance of the nodes. The structure is very similar to the *MaxAnswerTimePayload*.

```

enum { sec (1), millisec(2) } Unit;

struct {
    uint8          constMinPoaChangeIntPayload;    /* defined in 4.13.19 */
    uint8          Unit
    opaque          interval<0..2^16-1>;
} MinPoaChangeIntPayload

```

4.13.6 MaxUpdateIntPayload

The payload provides information about the idle timeout used for the M-SE connection. The node must send M-SE location update or authenticate packets maximal at this interval. Otherwise, the session will be deleted. Announcing this value helps to adjust the best update frequency for the connection update.

```

enum { sec (1), millisec(2) } Unit;

struct {
    uint8          constMaxUpdateIntPayload;        /* defined in 4.13.19 */
    uint8          Unit
    opaque          interval<0..2^16-1>;
} MaxUpdateIntPayload

```

4.13.7 SessionLifetimePayload

The payload gives information on the M-SE session lifetime. The *LifetimeUnit* can be seconds, MBytes or KBytes. Multiple payloads can be combined.

```

enum { sec (1), KBytes(2), MBytes(3) } LifetimeUnit;

struct {
    uint8          constSessionLifetimePayload      /* defined in 4.13.19 */
    uint8          LifetimeUnit
    opaque          values<0..2^16-1>;
} SessionLifetimePayload;

```

4.13.8 CertificateRequestPayload

The node requires a certificate issued by authority pointed in this payload. The payload contains the Distinguished Name of the Certification Authority, thus the certificate subject name. The encoding can only be DER in this payload type. If there are no preferences for Certification Authority, then the node can send an empty data value for *certificateDN*.

The Mobile Client and Mobile Server send separate *CertificateRequestPayloads* to each other and therefore, they may authenticate with certificates issued by different authorities.

```

enum { PKCS #7 (1), DER (2), ShaAndURL(3)} Encoding;

struct {
    uint8          constCertificateRequestPayload    /* defined in 4.13.19 */
    uint8          Encoding
    opaque         certificateDN[0..2^16-1];
} CertificateRequestPayload;

```

4.13.9 CertificatePayload

This payload carries the node's certificate. The encoding in the subtype can be DER, PKCS #7 (certificate chain) or URL. In the URL encoding, the node sends 20 octets SHA1 of the certificate followed by a URL link where the whole certificate can be loaded.

```

enum { PKCS #7 (1), DER (2), ShaAndURL(3)} Encoding;

struct {
    uint8          constCertificatePayload           /* defined in 4.13.19 */
    uint8          Encoding
    opaque         certificate[0..2^16-1];
} CertificatePayload;

```

4.13.10 ChallengePayload

The payload is used for password authentication very similar to PPP CHAP[7]. External AAA servers can be used in same way as by CHAP without any change of Radius protocol. The subtype is provided depending if the data is a challenge or response and calculated with MD5(default) or with SHA1.

```

enum { Challenge7 (1), ResponseMD5 (2), ResponseSHA(3)} DataType;

struct {
    uint8          constChallengePayload           /* defined in 4.13.19 */
    uint8          DataType;
    opaque         value[0..2^16-1];
} ChallengePayload;

```

4.13.11 EapPayload

```

struct {
    uint8          constEapPayload;                /* defined in 4.13.19 */
    uint8          0;
    opaque         eapValues[0..2^16-1];
} EapPayload;

```

M-KE specifies a container for EAP values as specified in [8]. Every type of EAP can be used, see 4.12.

4.13.12 PaddingPayload

```

struct {
    uint8          consPaddingPayload;             /* defined in 4.13.19 */
    uint8          0;
    opaque         padding[0..2^16-1];
} PaddingPayload;

```

The payload adds padding to the *EncrypredPayloads*, so it can be encrypted with block encryption algorithms, see 4.9.

4.13.13 IdPayload

```
enum { DN (1), UTF8(2), URI(3), FQDN(4) } IdType;

struct {
    uint8      constIdPayload      /* defined in 4.13.19 */
    uint8      IdType;
    opaque     value[0..2^16-1];
} IdPayload;
```

The ID type can be Distinguished Name (DN) in DER format, String in UTF8 string format, Fully Qualified Domain Name (FQDN) or Universal Resource Identifier (URI).

4.13.14 SessionIdPayload

This payload contains one session ID. The sub type is not used. The ID value is 4 bytes.

```
struct {
    uint8      constSessionId      /* defined in 4.13.19 */
    uint8      0;
    opaque     sessionID<0..2^32-1>;
} SessionIdPayload;
```

4.13.15 TrafficSelectorPayload

The traffic selector contains the transport parameter. The subtype *NetworkType* gives the type of the protocol. The *NetworkValues* contains the port ranges. If there is a single port then the start and stop values are equal. There are no port values for ICMP.

```
enum { TCP (1), UDP(2), ICMP(3), SCTP(3), all(4) } NetworkProtocol;

struct {
    select (NetworkProtocol) {
        case ICMP:
            /* no additional parameter */
        case TCP || UDP || SCTP
            uint16 startSourcePort;
            uint16 stopSourcePort;
            uint16 startDestPort;
            uint16 stopDestPort;
    }
} NetworkValues

struct {
    uint8      constTrafficSelectorPayload      /* defined in 4.13.19 */
    uint8      NetworkProtocol;
    opaque     NetworkValues;
} TrafficSelectorPayload;
```

4.13.16 HmacPayload

The *HmacPayload* contains protected hash values (HMAC) of all M-KE messages sent by the client and the server excluding the pending and retransmissions. The HMAC function is well known and it is used in IPsec and TLS/SSL and described in [12]. The HMAC value

is built with the hash type of MD5 or SHA1, specified in the *HashType* variable. The general structure:

```
enum { MD5 (1), SHA1(2), } HashType

struct {
    uint8          constHmacPayload      /* defined in 4.13.19 */
    uint8          HashType;
    opaque         hmac[0..2^32-1];
} HmacPayload;
```

The input text for the HMAC is a general concatenation of all significant packets:

```
Text = Node | ClientHello | ClientResponse1 | ClientResponse2 .... | ServerHello | ServerResponse1 |
      | ServerResponse2 ....
```

The Node string is equal to “1111” for the Mobile Client (MC) and “5555” for the Mobile Server (MS). The constants are set randomly, thus the only requirements are to be different at the MC and MS. The result is different HMAC values even when hashing the same messages at MC and MS. Mirroring attacks are avoided in this way. The retransmitted and pending messages are ignored and not considered. Retransmissions are result of packet losses or pending status. For the calculation of HMAC, all bytes of the fixed length HMAC payload are set to zero.

The EAP authentication can require more than 4 messages. In this case, all *ClientResponse* and *ServerResponse* messages are used in the HMAC building

The building of the HMAC depends on the authentication algorithm. The general principle is - if the sender has a private key then it signs the HMAC with it. The HMAC has null key in this case. If there is a shared secret or a Master Session Key (MSK) by the EAP, so it is used in the HMAC. The HMAC can be build with zero key, if no key is available. This is strongly undesired, because of man-in-the-middle attacks. The bytes of the key are repeated until the required size is reached if the HMAC key is less than the length of the hash function (SHA1 20 bytes MD5 16Byte). For example: if the key is 17 bytes and a 20 byte key is required then the first 3 bytes are added again at the end of the key, thus the result is one 20 byte key. The HMAC for the different authentication scenario are summarized to:

Case I

The Mobile Client authenticates with a password or EAP without MSK. The Mobile Server authenticates with digital signature (RSA or DSS). The HMAC by server and client is built with zero key (equal to zero, thus no key). The client sends the HMAC result as it is. The server signs its HMAC with its private key.

The HMAC sent by the server cannot be manipulated, since it is signed with private key. A man-in-the-middle attack will be noticed by the HMAC verification at the client. It contains all client and server messages. The server cannot detect manipulation, since the client’s HMAC is not protected. The trust lies on the client.

Using the password as a key by calculation of the HMAC is unsuitable. The verifier of the HMAC must know the password. Commonly, the password is unknown to the M-KE server. The password is verified by external AAA server. The M-KE server forwards the CHAP values for verification to the AAA server (radius). The AAA responds with accepted or rejected to the K-KE node. The AAA cannot currently verify the HMAC, since it is not standard Radius operation. The M-KE server cannot verify the HMAC without the password in clear.

Case II

The Mobile Nodes use shared secret or EAP with a derivation of the Master Session Key (MSK, see EAP). The client and the server use the MSK or the shared secret for a key in the HMAC function. The secrets are unknown to a potential man-in-the-middle attacker, thus no manipulation is possible. It must be stressed that the MSK value must be delivered with an authentication in EAP handshake.

Case III

The server authenticates himself with a password or EAP and the client with a digital signature (RSA or DSS). It is the inverse case of the first one. The HMAC is built from the input text with no key. The server sends the resulting HMAC as it is. The client signs the result with its private key.

Case IV

The client and server use passwords or EAP without a Master Session Key. This combination is highly undesirable, since there is no protection against a man-in-the-middle. If used, the HMAC is built without a key from the input text.

Case V

Both client and server use a signature. Then the HMAC is built without a key and signed with the private key of the sender.

4.13.17 NotificationPayload

```
enum { request (1), responce(2) } NotificationDirection;

struct {
    uint8          constNotificationPayload          /* defined in 4.13 */
    uint8          NotificationDirection;
    opaque         NotificationMessage[0..2^16-1];
} NotificationPayload;
```

The payload carries the notification *NoticationMessage* the structure of which is described in 4.10. The type of the message can be a request or response. The initiator sends a notification request and the responder replies.

4.13.18 Protection algorithms

The protection algorithms are coded in two bytes shown in Figure 4.3. The first four Most Significant Bits (MSB) define the authentication. The following 4 bits define the encryption method. The next two bits define the hash function used in the HMAC. The last significant bit defines the mode of operation native or tunnel mode (M in figure).

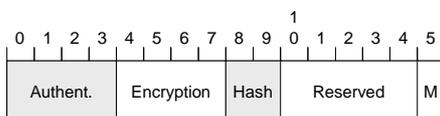


Figure 4.3: Protection algorithms

The following values are used:

Authentication Value	Algorithm	Standard
1	Digital signature RSA	PKCS1[5]
2	Digital signature DSS	DSS[6]
3	Password	M-KE

4	Shared secret	M-KE
5	EAP	EAP[8]
6	Null	

Encryption Value	Algorithm	RFC
1	DES CBC	RFC 2405
2	3DES CBC	RFC 2451
3	AES 128 CBC	RFC 3602
4	AES 192 CBC	RFC 3602
5	AES 256 CBC	RFC 3602
6	Null	

Hash Value	Algorithm	RFC
1	HMAC SHA	RFC 2104
2	HMAC MD5	RFC 2104

Mode (M) Value	Algorithm	Standard
1	Tunnel	M-SE
2	Native	M-SE

4.13.19 Payload constants

The type byte of the payload has the values:

<i>constDhValuePayload</i>	1
<i>constProtectionAlgPayload</i>	2
<i>constmaxAnswerTimePayload</i>	3
<i>constPendingPayload</i>	4
<i>constMinPoaChangeIntPayload</i>	5
<i>constMaxUpdateIntPayload</i>	6
<i>constCertificateRequestPayload</i>	7
<i>constCertificatePayload</i>	8
<i>constChallengePayload</i>	9
<i>constEapPayload</i>	10
<i>constPaddingPayload</i>	11
<i>constIdPayload</i>	12
<i>constSessionIdPayload</i>	13
<i>constTrafficSelectorPayload</i>	14
<i>constHmacPayload</i>	15
<i>constNotificationPayload</i>	16

The subtype bytes have values, which are defined in the payload definition, depending on the payload type, see chapter 4.8.

The *notificationType* in 4.10 has the possible values defined in the following table. The taken action depends on the local security policy. In the table are given some possible actions.

<i>Notification Type</i>	Meaning	Originator	Description
1	Server Discovery	client	The client requires a list of servers. The server should reply with its list.
2	Redirection to alternative server	server	The client must connect server from the list.
3	M-SE session is about to expire	client/server	The node is about the delete the current session. The node should start new M-KE.
4	Invalid M-SE	client/server	The M-SE session is invalid.
5	Node is going down	client/server	The node cannot serve this session anymore. The receiver must delete its sessions.
6	Node internal error	client/server	The node cannot serve this request due an internal error.
7	No proposal chosen	server	No proper proposal can be found.
8	Invalid signature	client/server	The node signature is invalid.
9	Revoked certificate	client/server	The send certificate is revoked.
10	Authentication failed	client/server	The credentials are invalid, i.e. password or shared secret.
11	DH parameter not supported	client/server	The DH group is not supported.
12	Decryption failed	client/server	The host cannot decrypt the payloads.
13	Server running on this client	client	The node also runs a server. The new server can be added in the list.

4.14 References in chapter 4

- [1] Dierks, T. and Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006
- [2] Mogul, J., and S. Deering, S., "Path MTU Discovery", RFC 1191, DECWRL, Stanford University, November 1990
- [3] W. Stallings, "Cryptography and network security", Forth Edition, Persion Edication, Inc 2006
- [4] Rivest, R., Shamir, A., and Adleman, L., "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM, v. 21, n. 2, February 1978.
- [5] Jonsson, J. and B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", RFC 3447, February 2003.
- [6] NIST, "Digital Signature Standard", FIPS 186, National Institute of Standards and Technology, U.S. Department of Commerce, May, 1994.
- [7] Simpson, W., "PPP Challenge Handshake Authentication Protocol (CHAP)", RFC 1994, August 1996.
- [8] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowitz, "Extensible Authentication Protocol (EAP)", RFC 3748, June 2004.
- [9] Haverinen, H, Salowey, J. "Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM)", RFC 4186, January 2006
- [10] Orman, H., "The OAKLEY Key Determination Protocol", RFC 2412, November 1998.
- [11] Kivinen, T. and M. Kojo, "More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)", RFC 3526, May 2003.
- [12] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.
- [13] Kaufman, C., Ed., "The Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005
- [17] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.
- [15] IANA assignet protocols numbers, <http://www.iana.org/assignments/protocol-numbers>
- [16] Zeilenga, K., Ed., "Lightweight Directory Access Protocol (LDAP): String Representation of Distinguished Names", RFC 4514, June 2006.
- [14] Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", RFC 3986, January 2005
- [18] Arkko, J. and H. Haverinen, "Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)", RFC 4187, January 2006.
- [19] Simon, et al., "EAP-TLS Authentication Protocol", RFC 5216, March 2008
- [20] Bersani, et al., "EAP-PSA Authentication Protocol", RFC 4764, January 2007
- [21] Cam-Winget, N., et al., "The Flexible Authentication via Secure Tunneling Extensible Authentication Protocol Method (EAP-FAST)", Work in Progress, October 2006.

5 Mobile Session Encapsulation

The Mobile Session Encapsulation (M-SE) is a protocol for data protection between the Mobile Client and Mobile Server. The M-SE packets are encrypted, contain authenticated integrity check, replay protected and have session hiding mechanism. The M-SE requires pre-configured keys, algorithms etc, which are delivered by the M-KE described in chapters 3 and 4. The M-KE negotiation must be successfully finished before every M-SE session. The M-SE runs over UDP and uses the same ports as M-KE (see 3.4).

The M-SE introduces a new approach for session hiding described in 3.7. There are novel methods for location update, see 5.8.1 and Mappers described in 5.1.1. The M-SE implements database SAM-DB for connection management described in 3.9. The well-established crypto methods, like 3DES, AES etc, are used in the protocol. The M-SE does not intent to use new crypto algorithms then to use the well-studied and implemented algorithms.

The M-SE delivers transparent network layer (UDP/TCP) to the end nodes. Transparent means that it is transmitted without any manipulation. The transport layer (IP) is constant during one session, but it is not transparent. The IP is constant equal to the predefined Mappers as described in 3.6.1. The public IP values change by the host movements and therefore, they are masked with Mappers.

The M-SE session is not tied to constant PoA parameter. The received packets are matched to the certain session using dynamic *SessionID* filed in the M-SE header, thus the received packet is processed independent from the PoA. When sending packet, the last known PoA (last received authenticated packet) is used by the Mobile Node. The current PoA parameters are stored in the SAM-DB and updated with every authenticated packet received (see 3.9). If there is no permanent traffic, the node should send updates to announce its current location. The update procedure uses notification packets (see 5.8.1). The time intervals for update execution is subject of the Mobile Location Update (M-LU) protocol, described in 7.

5.1 Tunnel mode

The M-SE operates in two modes: tunnel and native described in 3.6. The M-SE does not provide a tunnel, but it integrates existing external tunnels such as L2TP. The tunnel mode is typically used for building IP layer connections known as VPN. Link Layer connection are also possible but not discussed here. The layer structure in tunnel mode with L2TP or GRE is shown in Figure 5.1. How the tunnel works is not a subject of M-VPN.

The possibility for the physical separation of the Tunnel Node and Mobile Node provides a big advantage for the M-VPN. The Tunnel Nodes are usually already part of the provider infrastructure using AAA servers etc. The synergy effect is that the existing infrastructure can be used in M-VPN.

5.1.1 Mappers in tunnel mode

The Mappers mask (exchange) the public IP addresses of the Mobile Nodes, which change during the session see 3.6.1. The Mappers are constant during the M-SE session and are locally defined. The M-SE delivers constant and not transparent IP layer (transport). This is sufficient for building a tunnel over M-SE.

The Mappers are used for transport between the Tunnel Node and Mobile Node, thus they only have local significance. They are not announced between the Mobile Nodes in M-VPN. The Mappers on the server and client side can overlap, since there is no relation between

them. The addresses must be unique in the local environment. Private addresses can be used also.

There are two Mappers for every M-SE session: termination and session Mapper, described in chapter 3.6.1. The assignment of the Mapper depends on the node's ID and can be static IP or from IP pool. It is important to stress that the user ID must be unique. Otherwise the Mapper can overlap, which leads to routing problems.

The L2TP connection is initiated from the client (L2TP LAC) mostly. The Mobile Server may assign session Mappers from the IP pool. The advantage in this case is that the client ID must not be known in advance. The server chooses unique session Mapper from the IP pool. L2TP is considered more suitable for M-VPNs, since the client initiates the connection to the server. This fits well in the unidirectional nature of NAPT. The static Mappers are used by GRE tunnel, since the Tunnel Node points are fixed for these two clients.

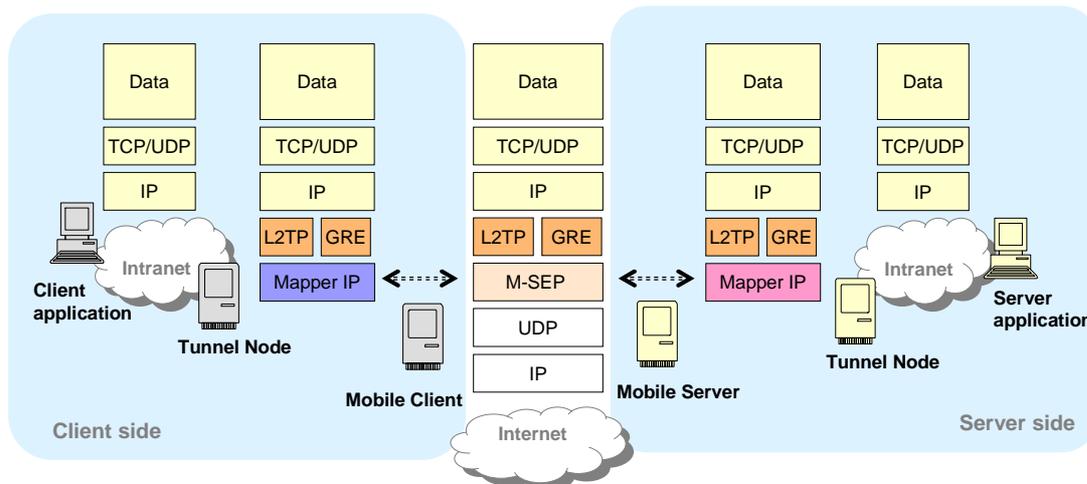


Figure 5.1: Layer structure of M-SE in tunnel mode

The termination Mapper defines the IP of the Tunnel Node. Setting the IP depending on the user gives the flexibility of using multiple Tunnel Nodes for one Mobile Server. Load balancing and administrative separation can be achieved in this way. For example: the nodes can be separated in different administrative groups and use different Tunnel Nodes. Figure 5.2 shows the distribution of to 3 Tunnel Nodes for administrators, users and guests.

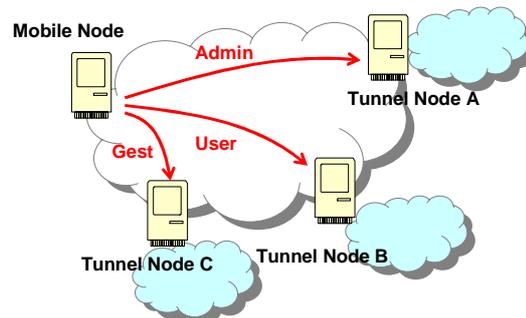


Figure 5.2: Mobile node with multiple nodes servers

5.1.2 Authentication in the Tunnel Node

Authentication and authorisation can be carried out at different layers in tunnel mode, since PPP, L2TP and M-KE support authentication. If the Tunnel Node is already deployed, the existing authentication processes could be used. Before implementing the authentication must be considered that, the different layer authentication can have an impact on security as described in 6.1

5.2 Native mode

The native mode is simplified deployment without tunnel. It can be partially compared to IPSec in transport mode where TCP/UDP communication can be protected. The difference is that the IPSec host is at the same time the end host of the connection, for example HTTP session. In M-SE, the end host can be different from the Mobile Server, which gives additional flexibility of M-SE comparatively to IPSec. The layer structure is shown in Figure 5.3.

The Mappers are used between the application and the Mobile Server. The applications are aware of the Mappers, which are part of the Intranet. This must be considered carefully for all bundled sessions and protocols carrying network parameters in their payloads. The Mobile Node does not perform any translation of the payload and therefore, bundled sessions may not work properly.

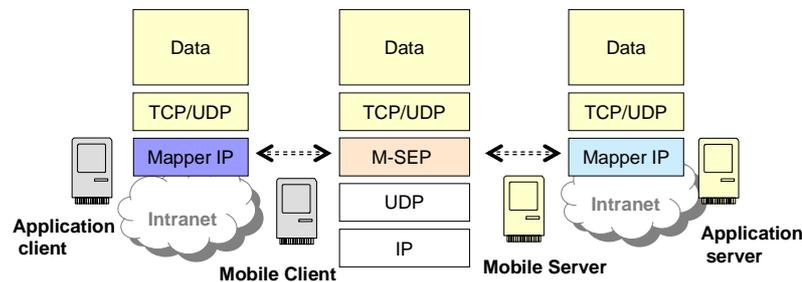


Figure 5.3: M-SE in transport mode

5.3 Packet structure

The packet structure according the notation in [1] is following:

```

struct {
    ClearMseHeader      clearHeader;
    EncryptedData      data;
} MobileSe

struct {
    ProtocolVersion    mobileSeVersion;
    Length             len;
    opaque             sessionId1[4];
    opaque             hmac[12];
} ClearMseHeader

struct {
    EncryptedMseHeader encrHeader;
    opaque             dataApplication [0..2^16-1];
    opaque             paddig [0..2^16-1];
} EncryptedData;

struc {
    opaque             sessionId2[4];
    uint16            sequeceNumber;
    uint8             notification;
    opaque            notificationDescription[0.255];
    byte              nextProtocol;
} EncryptedMseHeader

struc {
    uint8             version;          /* major | minor */
} ProtocolVersion

```

```

    struct {
        uint8      fragment;
        uint16     packetLength;
    } Length

```

The packet contains encrypted and unencrypted part. Both of them have a header: *ClearMseHeader* and *EncryptedMseHeader*. The unencrypted header is used for matching the M-SE session and integrity check. The encrypted header carries sequence number of the packet, advertised *sessionID* etc.

The *ClearMseHeader* header starts with the byte *ProtocolVersion*. The first 4 most significant bits (MSB) are the major version. The major version can be from 8 to 15. The version values do not overlap with the M-KE as described in 3.4. The 4 Last Significant Bits (LSB) contain the minor version, which can take values from 0 to 15.

The *Length* structure contains the length of the datagram and optional information about the fragmentation. The values are set in the same way as described in the M-KE specification, see chapter 4. The fragmentation is done over the encrypted data. The fragments have unencrypted M-SE header and encrypted chunk. Generally, fragmentation of the datagram at M-SE layer must be avoided through reducing the MTU size of the virtual interface.

The *SessionId1* contains of 4 octets giving the M-SE identification. The value is negotiated in the M-KE phase and can be changed during the M-SE session see 3.7. The HMAC value is used for integrity protection and authentication of the packet. The vector contains the first 12 bytes of the HMAC result based on MD5 or SHA1 as defined in [2]. The HMAC algorithm is part of the M-KE negotiation. The HMAC key is the *Sa_c* for MC and *Sa_s* for MS, see 4.6. The HMAC value is built from the whole M-SE datagram without the UDP and IP header. The HMAC field is set to zero in the calculation. The *data* is encrypted with the *Se_c* for MC and *Se_s* for MS. The encryption algorithm is negotiated in the M-KE.

The encrypted header (*EncryptedMseHeader*) is the first structure in the encrypted part. The first 4 octets are the announced session ID (*SessionId2*) see 3.7. The *sequenceNumber* gives the sequence number of the packet described in 5.5.

The *notification* byte gives the possibility to send embedded information to peer. The *notification* does not require a supplementary packet in this way. The notification byte has the structure shown in Figure 5.4. The MSB (Most Significant Bit) bit

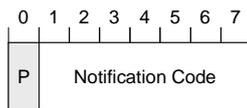


Figure 5.4: Notification byte

is the P flag. The flag gives the feedback from the sender's perspective, if the sender's PoA has changed. The flag is inverted when the PoA differs between two following received M-SE packets, see 5.8.1. The information is used in the location update procedure (5.8.1). The *Notification Code* is the 7 LSB (Least Significant Bits) in Figure 5.4. The zero means no notification. Every value greater than 0 is a notification. The codes are listed in 5.8. The *notificationDescription* gives a verbal description of the notification. The text is optional and can be used to add some additional details for debugging. The *nextProtocols* indicates the protocols carried in the M-SE. The protocol numbers are defined in [3]. The actual data is following in the *dataApplication* vector. The last vector *padding* adds a variable number of zeros to the data, so it becomes modulo of block size see 5.7.

The M-SE adds a variable overhead depending on the added padding and notification text. The minimum length starts with 30 bytes (240 bit). The size can be optimised in the future versions of this protocol. In this research document, it is not part of the focus. The usable data of interface with 1500 bit MTU size and IPv4 is 1500 – 96 (IP) – 64 (UDP) – 240(M-SE) = 1100 bits.

5.4 Packet processing

5.4.1 Inbound

When a packet is received by the M-SE process, the following steps are proceeded:

- The header structure and the version are verified.
- Using the 64 bit *SessionId1* value, the M-SE session is matched in the SAM-DB. The M-SE session in SAM-DB defines the parameters for decryption, authentication etc. If the M-SE session cannot be found, the packet is dropped and an optional M-KE notification may be sent see 5.8.
- If datagram was fragmented then the procedure is repeated until the hole datagram is assembled.
- The HMAC value is checked. The packet is ignored if the proof fails.
- The encrypted part is decrypted. If the decryption fails then the packet is dropped.
- The sequence number is verified according to 5.5. If check fails then the packet is dropped. The current sequence number is updated if the check is passed.
- The second *SessionId* is updated in the SAM-DB. The new *SessionId* is used for sending packets to the node, thus outbound.
- The optional notification is considered. The notification values and possible actions are listed in 5.8.
- The source IP and UDP is updated in the SAM-DB (current PoA values). This value is used for the P flag settings (see 5.4.2).
- The encapsulated packet is extracted and compared to the traffic selectors. If the packet does not match, it is dropped and M-SE notification is send.
- Mapper is added to the data and the packet is forwarded by the routing process.

5.4.2 Outbound

When a packet is received for outbound processing at the Mobile Node:

- Using the Mapper one dedicated M-SE session is selected. The M-KE is initiated if there is no active M-SE session and the node is a client. The packet is dropped otherwise.
- The traffic selectors are used to verify if the packet should be sent through this connection. If the verification fails, the packet is dropped.
- The encrypted header is built in the following way:
 - The node sets the encrypted *SessionId*. It can be one already used or a new one. The new one can be generated by pseudo random algorithm, see 3.7. If a new *SessionId* is set then it is added in the database.
 - The sequence number is increased by one and set in the header.
 - The current destination IP and UDP are compared to the last used (last packet). If the values differ, the P flag is inverted see 5.8.1.
 - The notification value is set if required.

- The destination PoA is stored and used for building the P flag).
- The next protocol value is set.
- The data and padding is added to the structure, so the *EncryptedData* structure is built. The data is above the network layer (not including the network layer).
- Then the clear header is build:
 - The protocols version is set
 - Length is constructed and if necessary, the datagram is fragmented.
 - The HMAC value is calculated over the whole M-SE packet, with HMAC values set to zero.
- The packet is forwarded. The IP and UDP header of the current PoA is the target of the packet.

5.5 Anti replay protection

For the replay protection, the same method is used as in ESP [4]. Every node has a defined window size. This is the maximum disorder when receiving the packet, thus the maximum delay in packets. Out of sequence packets are result of path diversity in IP networks. Using M-SE in conjunction with Quality of Server (QoS) can lead also to a higher delay of sequence packets and therefore the windows size must be always carefully considered. For example: protecting VoIP traffic in the high priority queue and the best effort in FTP. The window size is expressed in packets and the typical recommended value is 64, but higher values can also be used. The value 64 is recommended since disorder of 64 is very unexpected practically. The same value is used also in the ESP [4] implementations.

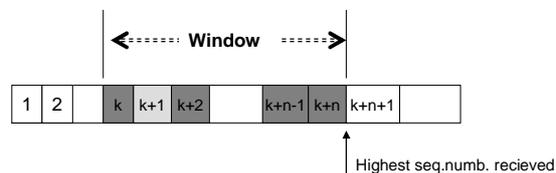


Figure 5.5: Window in anti replay protection

The highest received sequence number (authenticated packet) is in the right corner of the window shown in Figure 5.5. The left corner is shifted by the windows size. Every sequence number received within the window is marked. If one of the packets has been received already then the packet is dropped since it is considered as a replay attack. If the received number is out of the left side of the window, the packet is also dropped. More information of the method can be found in [4].

5.6 Unprotected notifications

Unauthenticated messages must be considered very carefully when they should have an influence on the M-SE and general on security protocols. This can be, for example: ICMP destination unreachable or fragmentation needed. More detail can be found in 6.2.

5.7 Traffic Flow Confidentiality Padding

Even knowing the size of the packet is a security issue. There are typical packet sizes for telnet, rtp etc. The size of the packet carried in M-SE can be masked using the greater

padding size. The padding is added to the length of block cipher, so the data can be processed by a block cipher. The sender can add a larger number of padding than the minimum needed. The size of the M-SE becomes significantly different to the original packet in this way. The sender can dynamically vary the size of the padding, so that no obvious correlation between the sizes can be found.

The larger packets increase the required bandwidth. Larger packets do decrease the performance of the Internet router since their performance is limited majorly to packets per second.

5.8 Notification

The notification is an integral part of every message, so no additional messages are required to send maintenance information. There are *Notification Code* bits giving the notification information, the values are from 0 to 127. The optional description gives detailed information, which can be used for debugging. The text part is not handled by the M-SE. The notification may require some response from the recipient. If there is no answer then the connection can be reset. The following table defines the notification values and expected actions:

<i>Notification Code</i>	Meaning	Reply	Description
1	M-SE session is about to expire	Yes, #3	The node is about to delete the current session. The receiver should start M-KE to re-establish a new session.
2	Invalid M-SE	No	The received M-SE packet is invalid.
3	Node is going down	Yes, #3	The node cannot serve this session any more. The receiver must erase its sessions.
4	Node internal error	No	The node cannot server this request due an internal error.
5	Revoked certificate	No	The sent certificate is revoked.
6	Authentication failed	No	The credentials are invalid.
7	Decryption failed	No	The node cannot decrypt the payloads.
8	Has a server running on this client	No	The node also runs a server. If the receiver has client functionality, it can connect and take the server parameter. Then the new server can be added to the list.
9	The traffic selectors are not matching	No	The client/server must close the session and start new M-KE and M-SE
101	Location update request	Yes, #102	Requires an immediate reply with empty M-SE packet (no data)
102	Location update reply	No	Reply of #101

5.8.1 Location update

Location update is a procedure for: (1) detecting if the PoA parameters have changed and (2) updating of the current PoA. When the PoA changes the nodes are not notified and all packets sent get lost see chapter 2. To overcome this problem, the nodes must send packets in certain intervals to update its PoA in this way. The location update in M-SE is done in two ways: Firstly, the PoA is updated in the SAM-DB with every received authenticated packet. Secondly, if there is no data to be sent the node can proactively send an update. The update is an empty M-SE packet containing notification number 101. The receiver must immediately reply with the notification bits set to 102.

All M-SE packets contain P flag. The flag is inverted every time when two following received packets of the same session have different PoA. The Mobile Client gets a feedback when its PoA has changed through received packets form the Mobile Server. Generally, this gives a feedback to the receiver, how its PoA changes from the perspective of the sender. An

example is shown in Figure 5.6. The PoA has changed at the 2nd, 6th and 8th received packet and the P flag is inverted. The send packets contain the P flag and give the information of the receiver.

The time points, when the P flag is inverted, are used to adjust the update interval of the receiver node. The time points are the input of the Mobile Location Update protocol (M-LU) see chapter 7. The inversion of the P flag is used instead of a single set to minimise the influence of the packet losses. If the information is carried in the single packet and it gets lost, then the estimation of the update time point will fail significantly. The inversion is more robust against the losses of single packets.

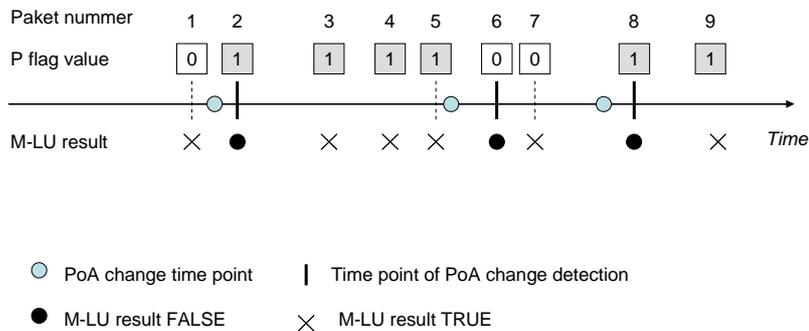


Figure 5.6: PoA change and location update

Every packet received is an execution of the location update, which can be initiated proactively through notification. The result of the location update is *true* or *false*. It is false, when the PoA has changed, thus the P flag has inverted. The result is true, if it has not as shown in Figure 5.6. The Boolean result is the input of the M-LU protocol describes in 7.

5.9 L2TP over M-SE

The Point to Point Protocol (PPP) [6] defines an encapsulation for transporting multi protocol packets across link layer. It is used at last mile in aDSL, UMTS (3G), ISDN, Dialup access networks. It provides client authentication with a variety of mechanisms such as PAP, CHAP and EAP etc. Furthermore, it is used to assign or to announce the different network parameters, like IP addresses, DNS servers etc.

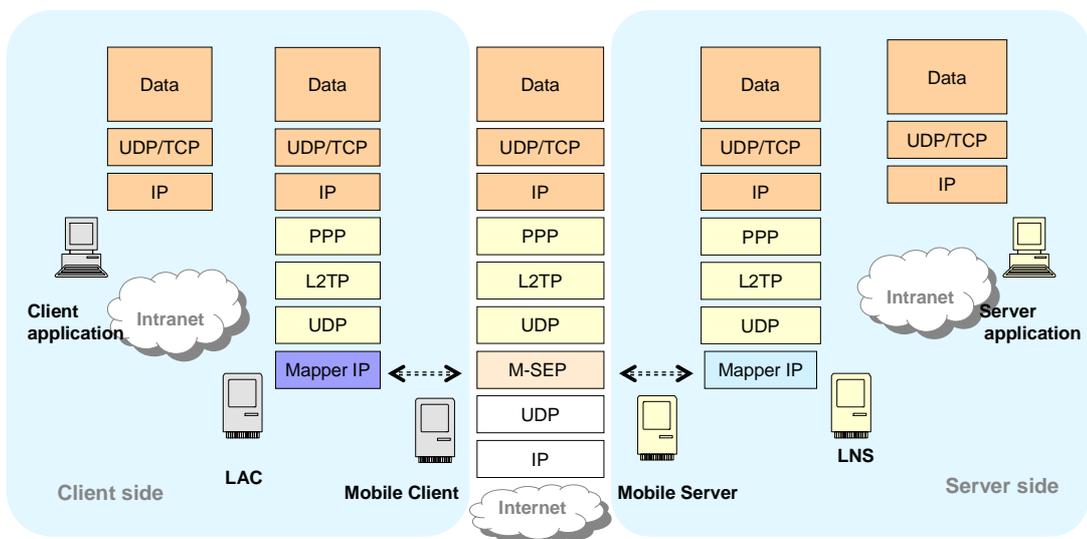


Figure 5.7: L2TP over M-SE

The Layer 2 Tunneling Protocol (L2TP) is a standard for transporting the PPP over packet switched networks (IP) and defined in [5]. The L2TP tunnel can carry over IP multiple PPP sessions and provides authentication mechanisms. The advantage of L2TP is the possibility of separating the physical link end-point from the PPP termination end-point. The traffic concentrators and aggregators can be used in the backbone structures. The PPP session can be forwarded into multiple instances. L2TP is deployed by the majority of the Internet Service Provider (ISPs).

The L2TP initiator is called LAC (L2TP Access Concentrator) and the terminator LNS (L2TP Network Server). The L2TP header is carried in UDP, which is transported by M-SE. Figure 5.7 shows the layer structure. The client node sends packets encapsulated in the PPP part of L2TP tunnel. The L2TP data is itself a tunnel in M-SE.

5.10 GRE over M-SE

General Packet Encapsulation (GRE) protocols defined in [7] is a tunneling protocol. It describes the encapsulation of a variety of protocols in each other simply using the GRE header. It is used for IP in IP encapsulation in most cases. The GRE over M-SE layer for IP encapsulation are shown in Figure 5.8.

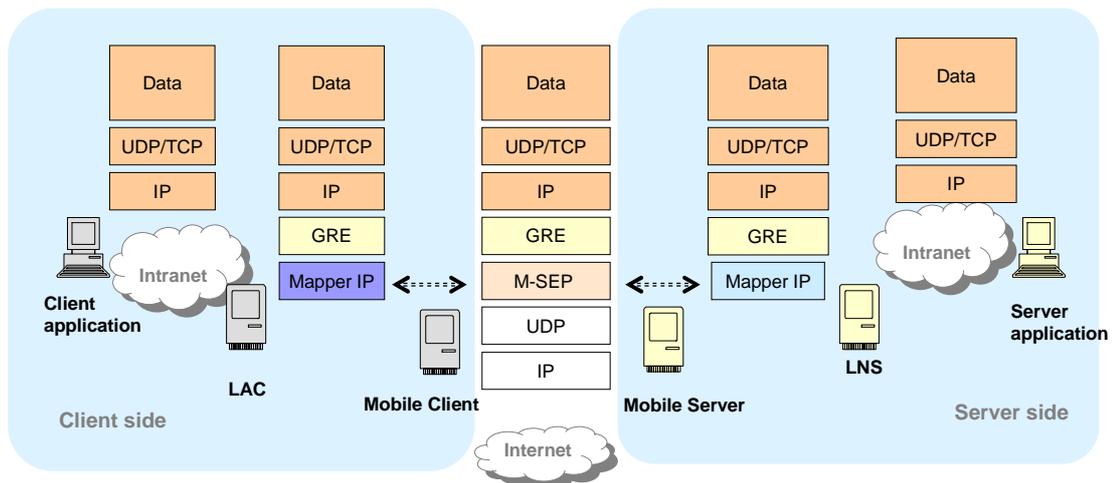


Figure 5.8: GRE over M-SE

5.11 References in chapter 5

- [1] Dierks, T. and Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006
- [2] Madson, C. and R. Glenn, "The Use of HMAC-SHA-1-96 within ESP and AH", RFC 2404, November 1998
- [3] IANA assigned protocol numbers, <http://www.iana.org/assignments/protocol-numbers>
- [4] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005
- [5] Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn, G. and B. Palter "Layer Two Tunneling Protocol 'L2TP'", RFC 2661, August 1999.
- [6] Simpson, W., "The Point-to-Point Protocol (PPP)", STD 51, RFC 1661, July 1994.
- [7] Farinacci, D., Li, T., Hanks, S., Meyer, D. and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, March 2000.

6 Security properties of M-VPN

The main topic of this chapter is the security properties of M-VPN. The suggested protocol must be carefully studied and analysed to minimise the risk of vulnerabilities and secure flows. The security of the cryptographic algorithms, such as Diffie-Hellman, AES, RSA etc are out of subject in the analysis. These algorithms are widely used in many protocols, like IPSec and SSL and it is assumed they are sufficiently secure from a mathematical viewpoint. Verbal analysis of M-VPN against certain attacks is made in this text. Furthermore, the state diagram is presented and deployment recommendations are carried out.

6.1 Authentication at different layers

The M-KE definition in 4.1 sets which authentication methods may be used by the Mobile Client and Server. This chapter, gives the motivation, why the usage is restricted exactly to these combinations.

The M-VPN is not starting from scratch, i.e. building new infrastructure. There is already an existing infrastructure and well-established protocols. Most deployments will use M-VPN in conjunction with already existing elements with PPP, Radius AAA etc. The good integration between all these protocols is the main requirements for building secure network. Generally, a big part of vulnerabilities is because of poor integration between the different applications. The application may work secure separately form each other, but the joint integration may suffer form vulnerabilities. The main problems are wrong assumption between the applications. For example using IPSec to protect SIP (VoIP). The IPSec session authenticates the hosts and not the SIP application. The SIP URI is not authorized and authenticate by the IPSec session, thus the SIP application may use bogus URI. The Integration of different security application must be analysed very carefully and it is target in this chapter.

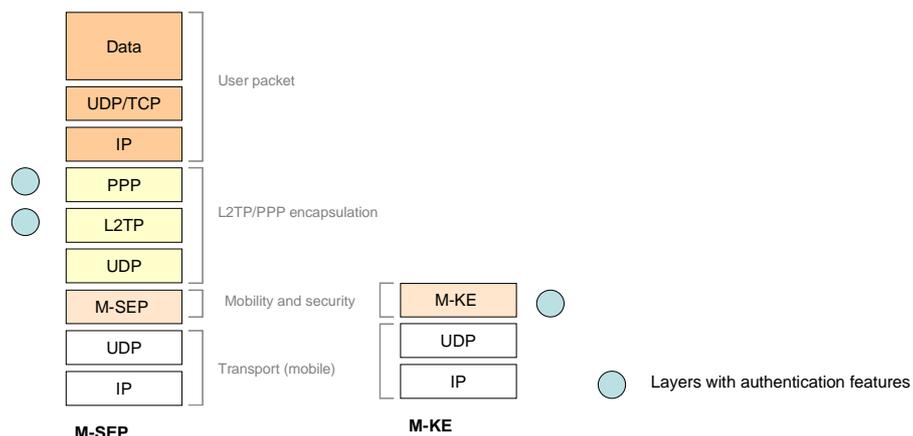


Figure 6.1: Layers L2TP and M-VPN

A typical layer structure of M-VPN with PPP is shown in Figure 6.1. There are generally three layers having authentication abilities, thus M-KE, L2TP and PPP, see Figure 6.2. The administrator has the possibility to set authentication at each of these three layers. It is a task of the security policy by the administrator to consider influences before switching features on or off. The problem is not limited to M-VPN, nut holds for all applications with overlapping functionality.

Authentication, authorisation and key generation in different layers can bring many problems for the real deployment. The first reason is matching of the ID between the layers.

For example: the x509 certificate used M-KE is issued to the user Alice, but the user authenticates himself as Bob in the following the PPP negotiation. Clearly, there is a mismatching of the ID and this can be an abuse from insider. It can be compared to a case when one person's driving license and the ID card are valid, but issued on different names. Mismatching of the ID's does not mean automatically an attack. This can be a legal case of inheriting administrative rights. In the last example, the certificate can authenticate the host and the PPP authenticates the user. This is obviously legal case.

First, there must be a clear policy of mapping of ID between the different layers. The layers are dependent from each other concerning the security. The interaction between the layers is not very easy, since the authentication methods can even have different syntaxes of the ID. For example: the PPP username is built typically as "username@domain", unfortunately the x501 structure has a different way of building the ID, for example "CN=user, OU=Arcor". It even does not allow "@" in the Distinguished Name according the x501 specification. The matching is a major problem and must be carefully considered in the local security policy. It cannot be defined globally in a standard, since there are different cases with different layer combinations.

The second potential problem arises when the authentication and the key generation are done at different layers. The Diffie-Hellman (DH) key exchange does not deliver authentication and therefore, it is vulnerable to man-in-the-middle attacks. The exchanged DH values must be authenticated for this reason. This is typically done by exchanging authenticated hash values. A problem is, for example, when the authentication is made in PPP layer and the DH exchange is made in M-KE. The DH is not authenticated, so the M-KE is not secure against man-in-the-middle. Therefore, the PPP connection over M-SE is unprotected against man-in-the-middle. The reason is that the PPP session is started only after successful M-KE negotiation and establishment of M-SE. It is not possible to use PPP authentication for protection of M-KE, since M-KE is finished before PPP begins (causality principle). There must be always authentication at M-KE level. If authentication is required in a PPP session, then there must be two authentications at M-KE and at PPP.

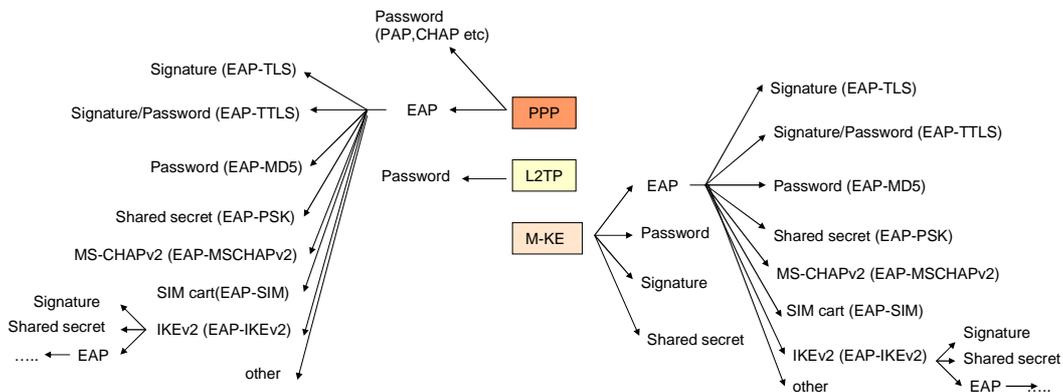


Figure 6.2: Authentication possibilities L2TP over M-SE

One possible solution in the example could be client authentication at PPP layer and server authentication at M-KE. The user PPP authentication can be made with a password. The M-KE server authenticates by certificate. The DH is authenticated by the server credentials (see 6.4.4). A manipulation can be noticed by the client because the server sends to the client a signed HMAC from all exchanges packets (containing also the DH public values). The server cannot notice any manipulation, since the client cannot build authenticated HMAC. The man-in-the-middle attack can be prevented only by the client. Naturally, the client has an interest of preventing manipulation. The exchange work in the same way as e-banking, thus server side certificate in TLS and client password in HTTP layer.

The complexity of the authentications combination on the different layers is underlined in Figure 6.2 and Figure 6.1. The different possible authentications are shown. There is a clear overlapping of authentication possibilities. For example: a password can be used in PPP, L2TP, M-KE, M-KE-EAP and PPP-EAP. There must be a clear policy what must be allowed and how the layers must interact. The L2TP, PPP and M-KE can be terminated in a physically different host, which makes the interaction between the layers almost impossible.

The authorisation is also an important point to be considered. It can be done depending on IP addresses, attributes in the username etc. In the most common, it will be at Tunnel Node or/and M-KE. The Tunnel Node can use the Mappers assigned by M-KE depending on the user ID.

M-VPN has an enormous flexibility for deployments of authentication. The authentication, authorisation and key exchanges must harmoniously co-exist and not lead to vulnerabilities. The local matching policy must precisely define the interactions and the matching rules. The local policy is depends on the deployment.

The M-KE defines that the integrated method for authentication must be preferred, thus signature, password, shared secret and password. If integrated methods do not provide the needed authentication, then EAP authentication may be used in the combination defined in 4.1.

6.2 Network security policy

The network security policy plays a very important role, but it is unfortunately often under-estimated. The security policy treats many aspects, like the interaction between the layers by the authentication and authorization (see 6.1). The network security policy is part of the security policy and it maintains the interaction between the network protocols. Two very important aspects of the network policy are described here.

The network security policy may be part of external node - Intrusion Detection System (IDS) [11, 10]. On the one hand, the separation of IDS and M-KE/M-SE server has the advantage that already existing IDS solutions [12] may be deployed fast. On the other hand, implementing security policy may require M-SE/M-KE layer information, which is encrypted and unavailable for the intermediate devices, like IDS. In this case, there must be an interface between the M-KE/M-SE server and IDS node.

6.2.1 Influence of unauthenticated messages on M-VPN

Special attention must be paid to the interaction between unauthenticated messages and security protocols (see 3.1). From a security perspective, the unauthenticated message must not influence the M-VPN. On one side, the unauthenticated message should be ignored. A typical example is the ICMP messages “host unreachable” or “fragmentation needed” [13], which influence the connections between two hosts. The notifications are untrustworthy, thus they can be bogus. An intermediate attacker can decrease the MTU size of the M-SE session by sending “fragmentation needed” notification. This leads to underperformance of the connection. The “host unreachable” sent by an attacker causes erasing of M-SE session according to the ICMP protocol [13].

On the other side, the unauthenticated messages cannot be ignored because of these security concerns. They are core mechanism in the global IP network. A connection drop may occur, if they are ignored. For example: if the MTU must be decreased in order to communicate, then ignoring the ICMP notification causes fatal disconnection. It must be emphasised that it is not realistic to require authentication and authorisation from all intermediate routers on the Internet.

The simple question “react or not react on received ICMP notification?” cannot be answered clearly and definitively. It depends on many factors and only indirect indices can give the answer with some probability. It is a task of the network security policy to define heuristic rules [14] for dealing with these types of messages. This is a problem limited not only to the M-VPN and it is problem for all security protocols, like as IPSec. The heuristic rules can be for example: revise check if the sender is replying to pings, trace route to the sender, allow a range of minimal MTU values, restrict the number of ICMP messages per second, etc.

6.2.2 Denial of Service attacks

The M-VPN secures the transported data, but does not provide any delivery guarantee. It is possible that the packets are dropped, wrongly delivered or retransmitted by the intermediate router. The packet routing and in general the transportation is not a protected service and an attacker can easily manipulate the IP and UDP header without even knowing the content.

Denial of Service attacks can be performed without any security knowledge, simply by replaying M-SE packets. The M-VPN gateway must decrypt the packet in order to verify the sequence number. This requires computational power of the gateway and an attacker can easily overload the gateway.

The risk of (D)DoS can be minimized by implementing intrusion detection and prevention policies with heuristic rules. They can be very simple, for example: if a host has sent more than 10 packets incorrectly, then it is blocked for 5 minutes. If the M-SE packet received was repeated more than 10 times, this M-SE session is blocked for 5 min. This is a heuristic solution using efficient simple rules for preventing most practical attacks. Every rule can affect trustful (innocent) M-SE sessions and this can become an undesired issue. The heuristic rules are trade-off filters for the bad and good connections using abnormal behaviour detection. This of course, does not prevent (D)DoD attacks, but my limit the effect of the attack.

6.3 State attacks

State attacks try to mislead the protocol in circular behaviour or to cancel certain session. The main instruments are out of sequence packets, replay or dropping packets.

6.3.1 Lost last acknowledgement

The security protocols are connection-oriented, because they negotiate security parameters. A general problem of all connection-oriented protocols is the so-called “lost last acknowledgement”. Both nodes can get in not corresponding states, so that the sent packets get lost. For example: Alice and Bob have one active session. After some time Alice sends finish/close message and deletes the session. If this finish message gets lost, Bob will not close its session and will keep using it. The sent packets by Bob get lost, since Alice has no session any more.

The not secure protocols can mange a workaround to this issue. The TCP [15] stack, for example, uses a simple rule: if a packet is received for not existing session, then a packet with control bit RST is send back. The receiver of the RST deletes its session. The connection is closed and both nodes arrive at the same state.

This problem is more complex in the security protocols, because they should ignore unauthenticated packets. The connection does not exist any more in the memory of the receiver and there is no key, ID etc. It cannot be answered in an authenticated way, consequently the peer will not trust this unauthenticated answer. Otherwise, an attacker can

reset every security connection. This problem cannot be solved principally by a design in any security protocols. It can be reduced only in the implementation using local policy with heuristic rules.

Partial solution in some security implementations, such as Cisco IPsec [9], is using “down-negotiation” state. After sending close notify, the session goes in state “down-negotiation”. The security parameters are kept for replying in an authenticated way if the last packet gets lost. The problem is only partially solved, because the time for keeping the state “down-negotiation” is limited. There is still the possibility to receive a packet after the “down-negotiation” is deleted. Furthermore, if the remote peer is not sending packets then its session will stay active. The session closure may be the trigger for backup actions or/and routing updates.

This is a security thread of all protocols and must be considered by the implementations. The M-KE implementation should implement the Cisco IKE strategy, since it limits to some degree the problem.

6.3.2 Rollback behaviour

Rollback behaviour occurs, when the send/receive timeout is shorter than the round trip delay. The host receives delayed packets with previous state and goes back to the last state (the state rolls back). If the transmission time is constant, this will cause delay in the negotiation by some packets, but the negotiation will be successfully finished.

The rollback becomes an issue when the transmission time is not constant during the negotiation. Some of the packets have bigger delay compared to other ones during the negotiation. The mobile environment uses mostly wireless connections, which are very unstable when the host moves. At some physical position during the movement, there is a strong signal and in some not. In areas with low signal, there will be more errors in the transmission. Therefore, some retransmission at wireless layer may be needed and the total delay will increase. When the host moves, there will be variable delay depending on the physical position of the node to the antenna. The variable delay in the negotiation can cancel the negotiation because of the retransmissions. Every protocol has a limit for retransmitting during the negotiation. When this number is reached, the negotiation is discarded. A simple example is show in Figure 6.3. Every state is retransmitted increasingly, which causes connection drop at the end.

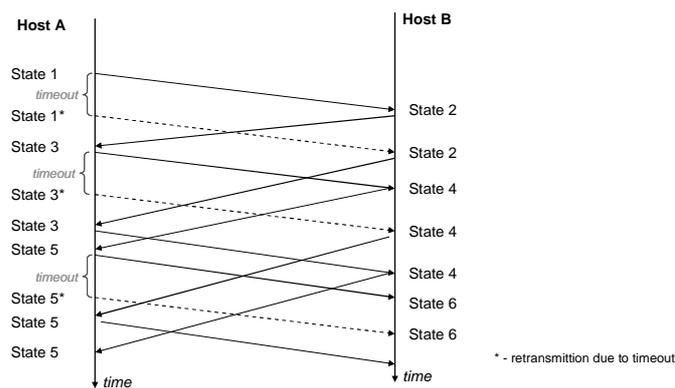


Figure 6.3: Rollback and retransmissions

The same behaviour can occur not only due to natural properties of the media, but also due to attack. An active attacker can disturb the commutation and cause circular behaviour of the both nodes. The problem cannot be solved principally, but can be reduced by using some rules. Dropped and out of sequence packets are expected in mobile environments. In order to reduce this circular behaviour, the following rules are set in M-VPN:

- No retransmissions of already finished states
- No rollback of any state is allowed
- Out of sequence packet are ignored

The comfort of the session decreases since some state differences will be noticed only after reaching a timeout. The states will be rest after the expiration of the timeout interval in M-KE. The values must be kept low for this reason. It is a trade-off between security and comfort.

6.4 Attacks on M-KE and M-SE

Attacks based on poor protocol design are very critical. They can be fixed only through new protocol version. The new version mostly will be not compatible with the old vulnerable one. Migration to a new protocol requires complex organisation. For instance, implementation issues can be solved on-the-fly by upgrading some modules. In this section, the M-KE and M-SE are analysed against some common design attacks.

6.4.1 Parallel session attacks

Two sessions are executed in parallel from a bogus insider in this attack. The target is to gain a different privilege level than the one assigned to the user. The attacker exchanges the IDs between the parallel sessions and can mislead the remote peer. The attack was first described by Abadi and Needham [1, 2]. It is briefly described here.

There is an insider (attacker) Malice, which shares legal secret K_{MT} with the authentication server (T). Alice (good user) shares the secret K_{AT} with the authentication server. The Alice will be attacked by Malice, thus Malice will be authenticate as Alice by Bob. Bob is the attacked node, which terminates the connections and uses the authentication server T. The basic exchange is shown in Figure 6.4. The messages are simplified for easy understanding. They can vary depending on the real protocol.

Malice starts two parallel negotiations with Bob. He is presenting himself as Malice in the first one and in the second as "Alice"¹. He has a legal password for the first negotiation as Malice. The attacker sends the ID of Malice and Alice in the first messages of the sessions 1 and 1'. Bob (attacked node) generated two nonce values one for Alice and one for Malice and returns them to the sender, message 2 and 2'.

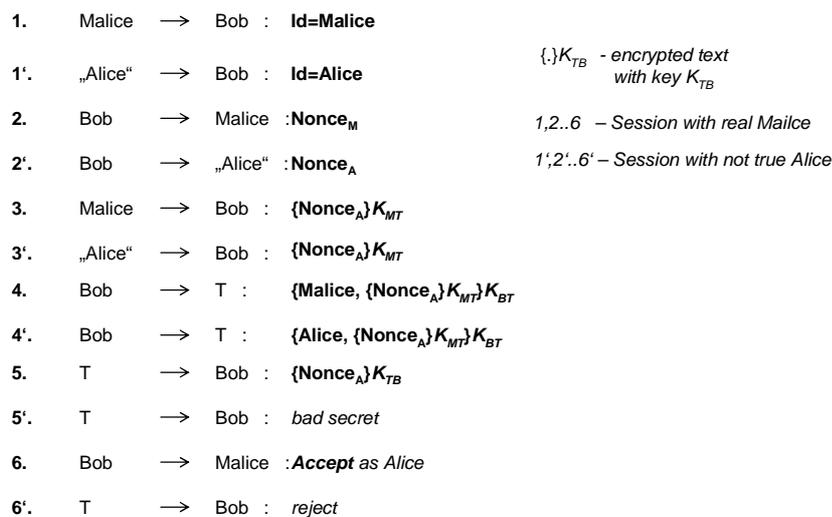


Figure 6.4: Parallel session attack

¹ "Alice" - quotations denote the bad Malice acting as Alice

The nonce values are tied to a specific session and are identification of it, like cookies. Malice discards the nonce dedicated to him (Nonce_M) and returns the Alice's Nonce_A in message 3. The 3' message is build as expected with the nonce of Alice. The nonces are encrypted with key K_{MT} , which Malice shares with the authentication server T. In message 4 and 4', Bob forwards the received message together with the ID to the T server for authentication. The messages are encrypted with the shared key K_{BT} between Bob and T, thus. The server T decrypts correctly the Nonce_A in step 5 using K_{MT} . The server returns the correct nonce. The T cannot properly decrypt the Nonce_A , since it not encrypted with the expected key K_{AT} . The first session is accepted in the following message 6, because the nonce is correct. The session of 6' is rejected. Bob accepts the first session and mapping it to Alice, which is actually the bogus Malice.

The core idea of this attack is using a nonce (cookie) for one session and performing authentication with any legal key for other session. The vulnerability is because the nonce (cookie) values are not bound to a certain ID in the authentication. The nonce is used for session and user identification. More information can be found in [2].

There are multiple protections in M-KE against this attack. There is an entry in the SAM-DB for every M-KE negotiation. The entry contains user ID, cookies, shared secret etc. The nonce is never the sole session identifier by the authentication. When signature is used for authentication, the ID is part of the certificate. The ID and the certificate are tied. When using external AAA in M-KE, then the ID and the password are authenticated for one dedicated session. The binding to one exact session ID is made by using request/response identifications in the radius requests. A parallel session attack is not possible in M-KE. Additional protection is that all sent and received messages including user ID, session ID, nonces etc are protected with HMAC against any sort of manipulation (see 4.13.16).

6.4.2 Reflection attacks

In reflection attacks [2], the message part is sent back to the originator. The target is to bluff the originator that this is the correct reply to its request and to accept it. In practical terms, the answer to the secret question is built on the question itself. Please notice that not the whole message is returned, but a part of it. The ID's, headers and other values may be manipulated.

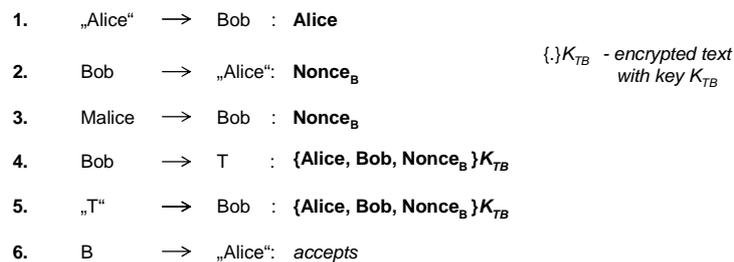


Figure 6.5: Reflection attack

The basic principle of the attack is shown in Figure 6.5. The 3rd and the 5th message are a reflection of the 2nd and the 4th message. The attacker Malice sends the nonce identifying Alice, at step 3. The attacker captures and reflects the authentication request sent by Bob to the authentication server at step 5. When the encryption is symmetric between T and B, then Bob accepts the session.

The client and server HMAC are built in different ways, containing different values for client and server in M-KE. The HMAC verification will fail in a reflection attack, since the server will receive HMAC from server type and it is expecting client's type, see 4.13.16. The reflection attack is not possible when using signatures, since the encryption and decryption key are different. A reflection of the server's signature to himself makes no sense and it will fail. Authentication with passwords use challenge response mechanism, where the authentication server matches the challenge to the password. Generally, reflection attacks are performed on symmetric encryption algorithms. They are impossible in M-KE, because of the different HMAC values.

6.4.3 Interleaving attacks

Interleaving attacks [2] consist of two or more negotiation runs. The principle is to collect sufficient number of values, like nonces/cookies, ID etc. This could enable to fake the ID. The interleaving attack is an overlapping use of a parallel session attack with reply attacks. Since M-KE and M-SE are resistant against parallel session attacks, the protocol should be resistant against interleaving attacks.

6.4.4 Man-in-the-middle

An intermediate attacker intercepts and manipulates the communication between the participants in a man-in-the-middle attack. This is a popular active attack, where the target is to bluff (mislead) and in this way to set or get the session key. After successful execution, the attacker can read and manipulate the exchanged data. More information can be obtained from [7].

To prevent a man-in-the-middle attack the key exchange must be authenticated. As already mentioned the Diffie-Hellman exchange does not provide any authentication and is naturally vulnerable to this attack. In M-KE, all exchanged messages are authenticated using the HMAC function. The client and server build HMAC values from all sent and received messages. The key of the HMAC depends on the authentication method see 4.1. If the hosts use EAP, then the EAP must deliver the master session secret used for the HMAC. If client and server have shared a secret then it is used for HMAC key. By authentication with signatures, the HMAC is build without a key. The HMAC itself is signed with the private key of the sender. When using the password at least one of the participants must use a signature. EAP authentication without delivering of authenticated master secret must be avoided. All M-SE messages are also protected by HMAC values with secret keys.

6.4.5 Replay attacks

One of the simplest attacks is the reply attack [7], thus an already sent packet is intercepted, copied and resent. The attack also aims to mislead the node to gain access for example. The attacker cannot decode the message, but it must know its purpose.

The protection of replay attacks in the key exchange (M-KE) is achieved through session unique cookies/nonces. These are protected by an authenticated integrity check. One of the nonce values is set by the client and the other by the server. If an attacker is replaying an old M-KE message, then the unique nonce/cookie will not match to the current session. The use of Hmac payload, see 4.13.16, protects the integrity of the messages and the cookies can not be exchanged without an notice. The receiver will drop the packet. Every negotiation in M-KE has a unique cookie and therefore no replays from previous negotiations are possible. If the attacker is using a packet from the current M-KE session these packet are ignored. These packets are treated as retransmission. There is not state roll back in M-KE as described in 6.3.

The replay protection in the M-SE is done by semi-unique sequence number. Each packet has an authenticated sequence number which must be received only once in a certain period.

The word semi is used because every digital counter is finite. The values will repeat in some long intervals. The packet is unique in period of 2^{16} packets in M-SE (this is the counter size). Theoretically, the packet can be replied in 2^{16} periods. Practically, the live time of the M-SE session is less than $2^{16}-1$ packets, so the session is no longer active. The maximum lifetime of a session is $2^{16}-1$ KB and each packet is ca. 1.5 KB. The attack cannot be performed in practical environment.

The replay attack can also be made with part of the message and not the whole packet. Authenticated integrity check is used in M-VPN. The M-KE and M-SE protection against replay attacks is quite similar to the well-studied IKE and ESP protocols.

6.4.6 Attacks due to type flow

The attacker uses vulnerabilities in the type flow of the protocol. The target is misinterpreting of the sent values and in this way to gain access. The idea is not to try to get the password but to replace it with some unsecure values. The best example is using certain encrypted nonce instead of encrypted session password, the values of which are known. Requirement for this type of attack is the same size and encryption of the key and the nonce. The attack was found on Neuman and Stubblebine protocol [4], see [2, 7] for more details.

Exchanging parts of encrypted messages in order to lead to misinterpreting is not possible in M-KE. All exchanged values are stored in payloads with different type identifiers. There is type and subtype field for every value giving the manner of interpreting the value. Furthermore, the datagram is protected by HMAC, so no parts can be exchanged. The payloads are interpreted because of their type and not because of their position.

6.4.7 Name omission

Every critical message of the protocol must be tied to unique session information. The unique session information is usually built of ID's, nonces and cookies. If this is not done then a name omission attack can take place. The attack was demonstrated on Needham-Schroeder protocol [5] and described in [6]. The attacker uses old sent values for demonstrating ownership of a private secret. For example: Alice and Malice establish a legal and correct session. Alice encrypts the session key with her private key and sends it to Malice. Malice can use this encrypted part to send it to Bob and in this way to show ownership of the private key of Alice. Bob cannot determine that this value is part of an old session not originally targeted to him. Malice knows the session key in the encrypted text, so it can start communication [2].

M-KE uses always Diffie-Hellman and in this way, the key material is unique for every session. Further, the HMAC protection of all messages protects against this problem.

6.4.8 Attacks using absence of integrity protection

The designers tended in the early versions of the security protocols to save on overhead by omitting the integrity check. Only authentication and encryption were used for protection of the traffic. This was a critical mistake opening possibilities for cut-and-paste or session-hijacking attacks. The attacker can exchange part of the encrypted message and the packet is forwarded to some bogus insider after decryption, see [7]. This attack was possible in ESP in encryption mode [7]. In M-KE and M-SE, every message with all values is protected by an authenticated integrity check HMAC.

6.4.9 Absence of semantic security protection

The aim of this attack is to extract some critical part of information, which is sufficient to get secret information. This is a general vulnerability meaning that the attacker does not need the whole messages decrypted and concentrates on certain parts. To prevent these sorts of

attacks in M-KE, Diffie-Hellman exchange is carried out and all payloads are encrypted. The protection costs more resources, since some of information may not be confidential. In the M-VPN design, the preference is not to save on computational power at the cost of security. The attacker can get “all or nothing”, thus all parts are encrypted with the key and algorithm.

6.4.10 Other attacks

Every day some new vulnerability is found and therefore, it is difficult to finalize it. The M-VPN is very close to TLS and IKE/ESP from a security perspective. These are well-analysed protocols with many deployments. Keeping the security design close to these protocols assures robustness.

6.5 Perfect Forward Secrecy

Perfect Forward Secrecy (PFS) is unfortunately a much-underestimated problem. Many modern security protocols are used without activation of this feature, like for example TLS, SSL, IPSec. (Currently most of the e-banking with SSL/TLS do not involve PFS). When PFS is not used, a simple offline attacks may be possible. For example: in SSL/TLS with server authentication RSA without PFS, the session key is generated by the client and encrypted with server’s public key. An attacker can record SSL/TLS sessions from certain server, which are point of interest. The private key of the server can be calculated offline even using simple brute force attacks. After some years, the private key will be known to the attacker. Then all recorded sessions can be decrypted. If the protected information in these old sessions is still relevant, then it can become a big problem. For this reason, in M-VPN a new key is generated via Diffie-Hellman for every session. All credentials, like ID’s, are also protected according PFS in M-KE.

6.6 Oracle services

Oracle services means that the gateway inadvertently provides cryptographic operations. Every security server or gateway in some way provides oracle services to a potential attacker. An attacker can send some messages and the gateway will try to decrypt them. It will fail in most cases, but the gateway may return values in some cases, which can help the attacker. It can be compared to typing some numbers in decryption machine and analysing what happens. A popular attack is the name omitting described in 6.4.7. Most of these attacks are on symmetric ciphers, where the same key is used for encryption and decryption. To limit this type of attack in M-KE, symmetric encryption is avoided, i.e. no clear text secrets are sent with symmetric encryption. General, the IDS limits the oracle services when dropping multiple tries from single host.

6.7 State diagram

The mobile environments characterize with packet losses, delays, out of sequence packets etc. These events lead to abnormal changes in the states. Having fewer states reflects in less complex structures, which are more suitable for the mobile environment. Therefore, the M-KE and the M-SE protocols are designed to have fewer states.

6.7.1 Server state diagram

The server’s state diagram is shown in Figure 6.6 using UML 2.0 presentation [8]. Every received packet by the server is mapped to: an existing M-KE, a new M-KE or an existing M-SE session. The mapping is made using header M-KE/M-SE values, like session ID, cookie

etc. The server can have multiple parallel negotiations and sessions. Every M-KE negotiation and M-SE session has its own set of values presented with grey background in Figure 6.6.

- A new M-KE negotiation is created when the server receives *ClientHello* (CH) with new cookie values. The first state of the M-KE negotiation is “Verify CH”, where CH is abbreviation for *ClientHello*. The packet structure is verified in this state, thus presence and format of all values. If the check fails, the negotiation is erased and notification may be sent. If the check is successfully passed, then the negotiation goes to “SH Update SAM-DB” (SH is an abbreviation for *ServerHello*). The values in the SAM-DB are updated and the *ServerHello* message is sent to the client (generation of DH values, algorithms etc.). The *ServerHello* can be pending notification or can include all data. The pending notification is sent if the server cannot generate the full message immediately (see 3.5). After successful update of the SAM-DB, the server returns to wait for new packets. There is no need to keep the last state at the server side, since the client message *ClientHello* or *ClientResponse* contains sufficient information to handle the packet. The server must only limit the negotiation in time.

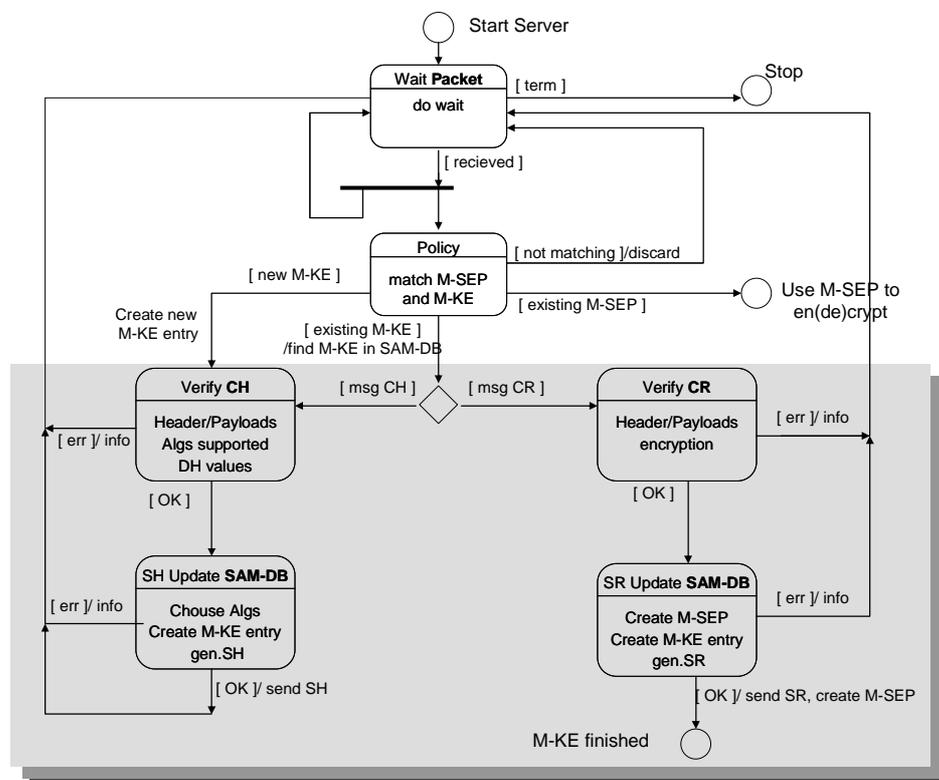


Figure 6.6: Server state diagram M-KE

- When the server receives packet to existing M-KE negotiation, it goes to verify the content in “Verify CR” or “Verify CH” (CR is abbreviation for *ClientResponse*). The server updates the SAM-DB and creates a *ServerResponse* or *ServerHello*. The message can contain pending notification or response with all information. If there is sufficient information, the M-SE entry can be created. After the M-SE is created, the M-KE negotiation is erased.
- If the received packet is M-SE, then the packet is decrypted according to the M-SE session values in the SAM-DB.

- The non M-SE and non M-KE packets are matched against the M-SP, if they must be protected with certain M-SE session

In order to handle multiple negotiations the server must fork to different processes by receiving messages. This is expressed by the first block “Wait Packet” in Figure 6.6.

The Figure 6.7 shows a simplified example of negotiation without any pending and retransmissions. The Server waits for packets (“Wait Packet”). After receiving a *ClientHello* (“Policy”), the MS verifies it (“Verify CH”) and sends the *ServerHello* message (“SH Update SAM-DB”). After receiving a *ClientResponse*, the MS verifies it (“Verify CR”) and generates *ServerResponse* (“SR Update SAM-DB”).

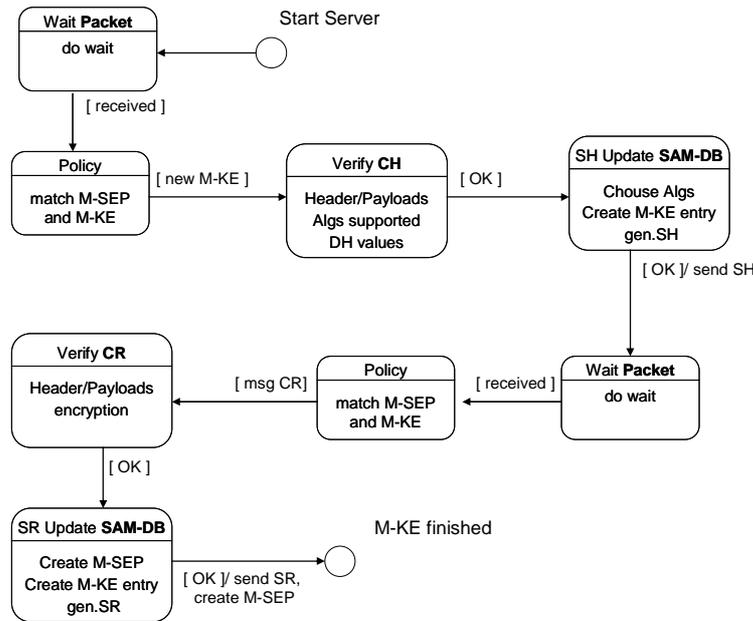


Figure 6.7: Example of MS state flow

6.7.2 Client state diagram

The state diagram of client is shown in Figure 6.8. The state diagram is very similar to the server diagram. The main differences to the server state diagram are because the client implements retransmission and pending. The server does not retransmit packets or pending (see 3.5). There can be multiple negotiations and therefore, the client must keep different states for every negotiation. The negotiation state is shown on the grey background in the Figure 6.8.

The client matches a received packet and performs the following actions:

- If the packet is non M-KE and non M-SE, and must be protected, then new M-KE negotiation is created in SAM-DB. In state “CH Update SAM-DB”, the *ClientHello* is prepared and sent to server. Then the client goes to state “Wait ServerHello”, where the client may retransmit the packet if no answer is received within certain time. The negotiation may be terminated by timeout too. If *ServerHello* message is received, the negotiation goes to “Verify SH”, where the message is verified. If the message contains pending notify, the client goes to state “Pending CH”, where after expiring the polling intervals the message is resent (state “CH Update SAM-DB”). If the received *ServerHello* contains all information, the client goes to state “CH Update SAM-DB”, where the client prepares and sends the *ClientResponse* message. After sending the *ClientResponse*, the node goes in state “Waiting

ServerResp”. When the *ServerResponse* is received, it is verified in state “Verify SR”. If the message is pending notification, the client waits in “Pending CR” for resending the message. If the *ServerResponse* contains all information, the client creates the M-SE session. The packets, which match the M-SP, are encrypted according this entry.

- If the client receives M-KE packet, it goes to the corresponding state of the negotiation.
- The M-SE packets are decrypted according to the M-SE entry.
- The packets matching to M-SP and are non M-SE/M-KE are encrypted with the M-SE entry linked the M-SP.

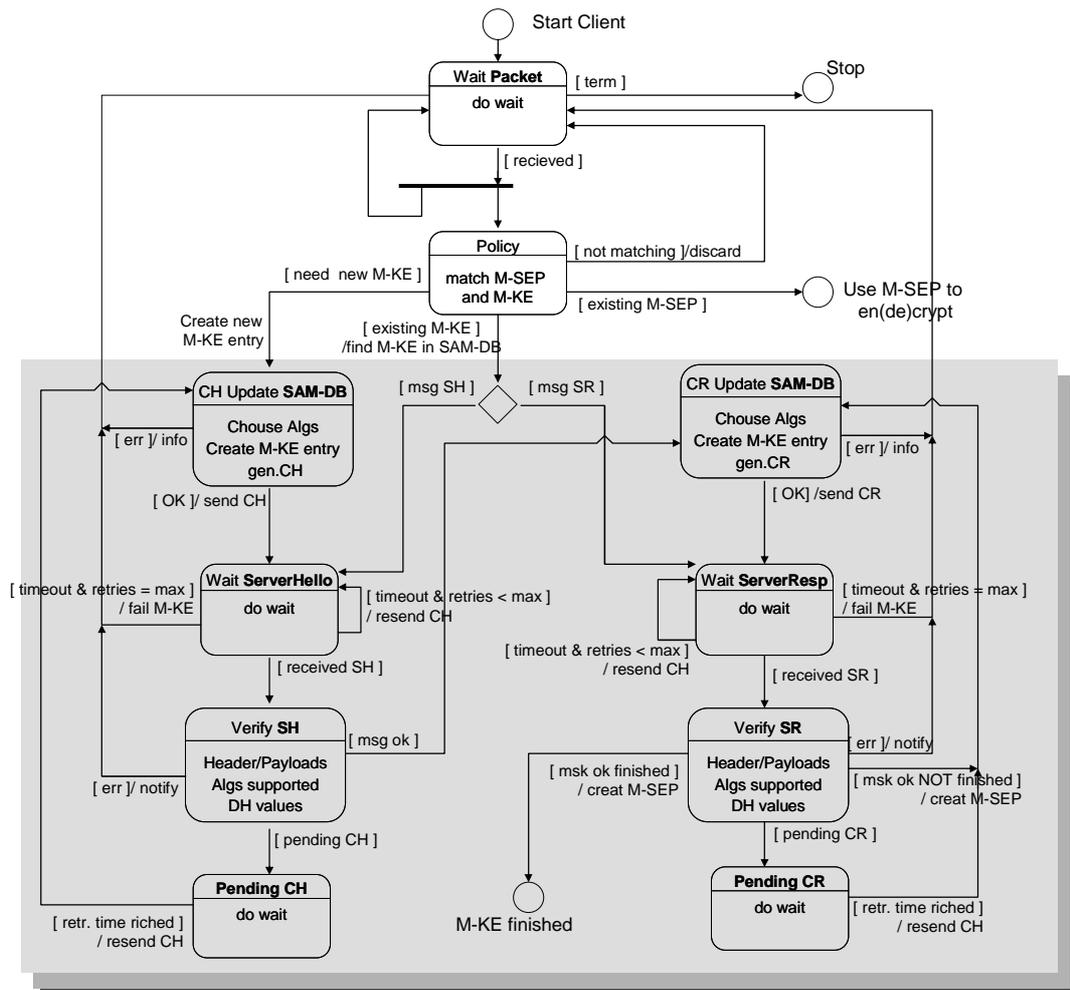


Figure 6.8: Client state diagram

A simple example is shown in Figure 6.9 without any pending and retransmissions. The MC wait for IP packet to be protected (“Wait Packet”, “Policy”). After receiving the packet, the MC sends *ClientHello* message (“CH Update SAM-DB”). The MC waits for *ServerHello* message (“Wait ServerHello”, “Policy”). The received packet is verified (“Verify SH”) and a *ClientResponse* is generated (“CR Update SAM-DB”). After receiving (“Wait ServerResponse”, “Policy”) and verification of *ServerResponse* (“Verify SR”) the M-SE session is created.

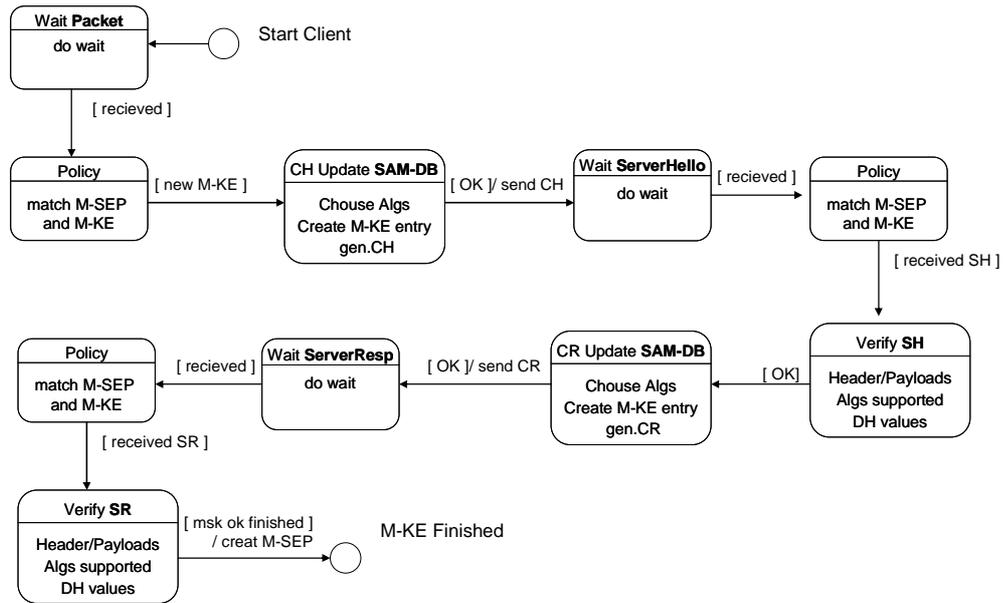


Figure 6.9: Example of MS state flow

6.8 Considerations regarding buffer overflows and injection attacks

Buffer overflows [16] are attacks based mostly on poor implementation of old C code. The basic idea of the attacks is to overwrite some memory segments using oversized variables. The overwritten segment can contain executable codes and in this way, an attacker can place code instead of the original one. If the programmer checks for the size of all variables, the overflow attack cannot happen. The most modern programming languages (Java, Net) have strict definition for the size of the variable and buffer overflows are not possible. The format must be checked strictly additionally for every variable to avoid injection code attacks [17].

6.9 Conclusion

The M-VPN is a strong protocol based on the experience of IPsec and SSL/TLS. The protocol transforms these security strategies in mobile environments. The security architecture is simplified to achieve clear design and fewer states. The M-VPN has no vulnerabilities to the known attacks, which are discussed in this chapter.

6.10 References in chapter 6

- [1] Abadi, Martín, Needham, Roger, "Prudent Engineering Practice for Cryptographic Protocols", IEEE Transactions on Software Engineering, 1995
- [2] Mao, Wenbo, "Modern Cryptography: Theory and Practice", Prentice Hall, 2006
- [3] Syverson, Paul F.: On Key Distribution Protocols for Repeated Authentication. Operating Systems Review 27(4): 24-30, 1993
- [4] Neuman, B. C. and Stubblebine, S. G., "A note on the use of timestamps as nonces, ACM SIGOPS", Operating Systems Review, 27(2), 10-14. 1993
- [5] Needham, Roger and Schroeder, Michael, "Using encryption for authentication in large networks of computers", Communications of the ACM, 21(12), 1978.
- [6] Lowe, Gavin. "An attack on the Needham-Schroeder public key authentication protocol. Information Processing Letters", 56(3):131--136, November 1995.
- [7] Stallings, William, "Cryptography and Network Security", Prentice Hall, 2006
- [8] "Unified Modeling Language", <http://www.omg.org/spec/UML/2.1.2/>, 11.2007
- [9] Cisco Corp, <http://www.cisco.com>
- [10] "The Science of Intrusion Detection System", Cisco Inc, 2004
- [11] Garuba, Moses; Liu, Chunmei; Fraites, Duane, "Intrusion Techniques: Comparative Study of Network Intrusion Detection Systems", IEEE, ITNG, April 2008
- [12] Snort, Network Intrusion Detection System, <http://www.snort.org/>
- [13] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, September 1981.
- [14] Zhou, A., Blustein, J., Zircir-Heywood, N., "Improving Intrusion Detection Systems Through Heuristic Evaluation", IEEE CCECE, May 2004
- [15] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.
- [16] Gerg, I., "An Overview and Example of the Buffer-Overflow Exploit". pps. 16-21, The Newsletter for Information Assurance Technology Professionals, 2005
- [17] Hope, Paco; Walther, Ben "Web Security Testing Cookbook", O'Reilly Media, Inc., 2008 p. 254

7 Mobile Location Update protocol

The Mobile Location Update (M-LU) protocol is a mathematical algorithm description rather than a technical specification. The protocol answers the question when to update the mobile host's PoA? (see the definitions in 2.2). In the ideal case, the mobile host must update its PoA at the communication peers instantaneous after a movement. Otherwise, the participants keep sending packets to vacant PoA that is not used anymore by the mobile host. Consequently, the sent packets get lost.

Unfortunately, the ideal target for immediate update after relocation cannot be fulfilled. The mobile host is practically unaware of its PoA as shown in 2.4 and 2.5. The host is not notified when its PoA changes and it cannot initiate an update. The application must proactively check if the PoA has changed for a certain remote peer. Depending on the result, some actions are carried out. The PoA can be different for the different peers (see 2.4), thus the PoA status must be checked for every connection.

Choosing the optimal update time point is a general problem not limited only to mobility. The same issue can be found in protocols, like VoIP (SIP), IKE, XMPP etc. The issue oscillates in mobile environments, where PoA changes are frequent. The current applications overcome this problem by performing updates at constant intervals and by assuming static environment. The receiver of the update notification extracts the sender's PoA from the IP and UDP headers. The sender does not know its PoA, but by sending an authenticated packet, it delivers the PoA in the header of the packet. Involving an expert knowledge of networks together with the application requirements, the update interval is set in advance. Commonly, short constant intervals from 10 to 90 sec are used in VoIP SIP applications for example.

The constant intervals are not adapted to the changing network conditions and movements of the mobile host. This leads to underperformance regarding the network load and CPU resources, as shown in 11.2.1. In a mobile environment, the problem leads to frequent packet losses and overload on narrowband links.

The main idea is to make the location updates proportional to the probability of a PoA change. During the time with high probability of PoA change, the updates must be frequent. The updates must be rare when there is a low probability of PoA change. In this way, network resources and packet losses can be minimized.

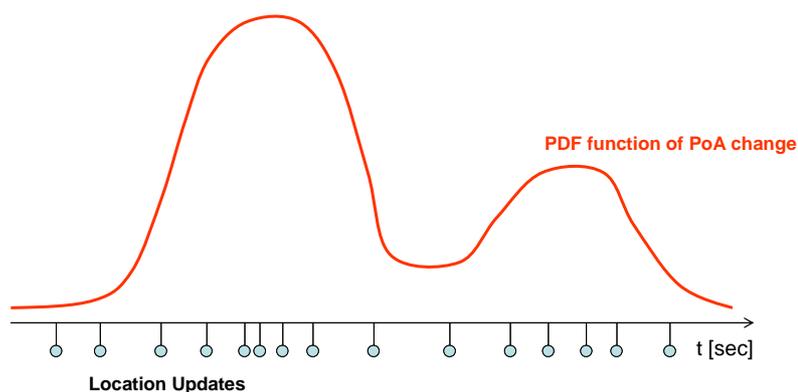


Figure 7.1: Updates proportional to PDF of PoA change

The Figure 7.1 shows the principle: high Probability Density Function (PDF) means more updates and vice versa. Let us take the example of a typical person working during the day. There is a high probability of PoA change in the rush hours. Therefore, the updates should be

frequent. At midnight there is less probability for PoA change and consequent less update are sufficient.

Considering the probability for PoA change helps to achieve much better results than using constant intervals. Better results mean less disconnection time with less network and CPU resources.

The main difficulties are: (1) how can classical mathematical algorithms be used for estimation and (2) how can the probability density function be constructed.

7.1 Challenges of the update procedure

The main challenge with M-LU is how to generate the Probability Density Function (PDF) of the PoA change. Every mobile device has different behaviour and a different PDF. The quality of the PDF is decisive for the performance of the algorithm. The key idea is to use the past changes to estimate the probability of a PoA change. The past values give quite a good estimation of the real PDF. The estimated function can be updated with every new occurrence of a PoA change.

The well-developed classical estimation methods cannot be deployed in a straightforward way. They require numerical values typically delivered by some measurement, like for example interval of 5 sec. The measurement consists commonly of signal with added noise. The target is to separate the noise of the signal and to obtain better estimation in this way. The update procedure in M-VPN delivers Boolean values and not numerical values as required. The Boolean values contain the information if the PoA has changed or not. The time point of the PoA change is unknown to the mobile host. There are not numerical measurements and therefore, the classical estimation method cannot be used directly. The time point of PoA change is not measurable in sense of estimation theory. The time can be narrowed down to the update interval. This is the time interval between two following PoA updates. For example: if the update interval is 300 seconds, then the disconnection occurs somewhere within these 300 sec.

Decreasing the update interval improves the estimation's precision of the PoA's change time point on the one hand. On the other hand, it increases the resources expressed in update packets and CPU utilization. There is clear trade off between the intervals size and the resources.

The reason of the update dilemma is that update procedure consumes resources itself, which must be minimised. Expressed in classical terms, the measurement is in the same time setting of the signal. It can be compared to calling the information desk of the railway station and asking if the train has passed. It becomes quite intensive task to detach the arrival time when getting answers only "yes or no", since the phone call costs money. The simple question is how to minimise the telephone costs and in the same time understand when the train is at the platform with sufficient precision.

7.2 Solution methods

Three novel frameworks are developed in this thesis. They estimate the PDF and calculate the update time points.

The frameworks represent general concepts for solving the problem, thus using statistics, analytical and Fuzzy method. Understanding and evaluating these three strategic possibilities gives objective comparison between the methods. The choice of optimal method depends on the implementation scenario, like mobile phones, cars, slowly moving vehicle etc. The methods have different properties and perform differently depending on the scenario. The frameworks are based on the following estimation methods:

- Stochastic solution using Sequential Monte Carlo is the first method presented in chapter 8. Gathering sufficient number of past PoA change events helps to construct the PDF, regardless of whether the system model is linear or not. The method represents a solution of the Bayesian equation using the statistics of past events. The method is known as Particle filter. It is expected that this method performs well in complex moving scenarios (complex PDFs), which cannot be approximated to same known curve, like e.g. Gaussian distribution curve.
- Adaptive Fuzzy controller using expert knowledge and training methods is the second method described in chapter 9. Fuzzy logic deals with subjective knowledge equivalent to verbal descriptions with multivalent values. The advantage of Fuzzy logic is the ability to use expert knowledge descriptions of the nodes movements. The experts can pre-define some raw rules to provide the main properties of the Fuzzy controller, like for example the weekly cycle of an ordinary employee. The training of the controller is made with the help of One Pass (OP) method. The rules are optimised with Recursive Least Square (RLS) method. This method is expected to be good for patterns, which change in time, in the time in combination with some predefined knowledge, like the day cycle of worker.
- Analytical solution based on extended Kalman filter is the last method developed in chapter 10. The famous Kalman filter [1] solves the Wiener problem [3] analytically and in this sense is optimal. Unfortunately, the Kalman filter can be applied only to linear and Gaussian models. Linear models cannot describe the movement of Mobile Node. A common solution presents the extended Kalman filter (EKF) [3], which approximates a non-linear system by a linear one. This is not an optimal solution as the Kalman filter, but achieves very good results. The EKF is used in the context of M-LU for estimating the next PoA change time point. The update times are calculated using here defined transformation functions. The method represents the classical analytical solution. This method is expected to be good for slowly changing patterns without any previous knowledge, like train movement.

7.3 Abstraction model and terminology

This chapter provides terminology definitions and abstractions before going deeper into method presentation.

The location update denotes the procedure of updating the PoA at the communication peers (see 2.2). In the context of M-LU, it returns also feedback to the mobile host if its PoA has changed. It consists of request and response packet for this reason. The request packet informs the receiver of the new PoA and the response returns the feedback to the sender. The feedback is called return code of location update procedure. The procedure is part of the M-SE protocol and defined in 5.8.1.

The return code of the update procedure is Boolean – true or false. The “true” means that the PoA is the same in the last and the current execution of the procedure. The PoA has not changed in this interval. The return code “false” means that the PoA has changed between the last and the current update. The PoA is unknown to the participants between the PoA change time point and the following update. This interval is called the Disconnections Interval (DI). All sent packets get lost in the Disconnection Interval since the participant send the packets to

an unused PoA. The M-LU concentrates on the prediction of the “false” events, thus changes of PoA.

Definitions:

Update Time Point (UTP) is the time point of performing the update procedure. The update procedure does not have zero execution time. We consider the point, when the Boolean result is received.

Event Time Point (ETP) is the time point, at which the host changes its PoA (moves).

The filters try to predict the coming Event Time Point (ETP), thus prior estimation of ETP. The prior estimation and prediction are synonym in this text.

The filter consists of two stages: prediction and update. They are executed in cycles following each other. In the prediction phase, the prior estimate is made. After the measurement becomes available, the coefficients of the filter are updated, thus posterior estimation is used. Each cycle begins with setting the Update Time Points and their execution point. After the PoA change occurs, the cycle begins again. The new cycle begins from the first Update Time Point (UTP) after the Event Time Point (ETP).

In general, to work with absolute time values is very inconvenient. The values increase constantly and patterns are difficult to compare. A much better solution is to work with intervals between critical points. The points of interest in the filter cycle are set relative to the beginning of the cycle, thus to the Update Time Point (UTP) after the Event Time Point (ETP). This is a very important conclusion for the further works with filters in this thesis. Estimation of the probability function is made always in one filter cycle. It must be noticed that the filter cycle cannot begin with the ETP since it is not measurable as already underlined in the previous chapter. The newly defined points and intervals are show in Figure 7.2 and their definitions are following.

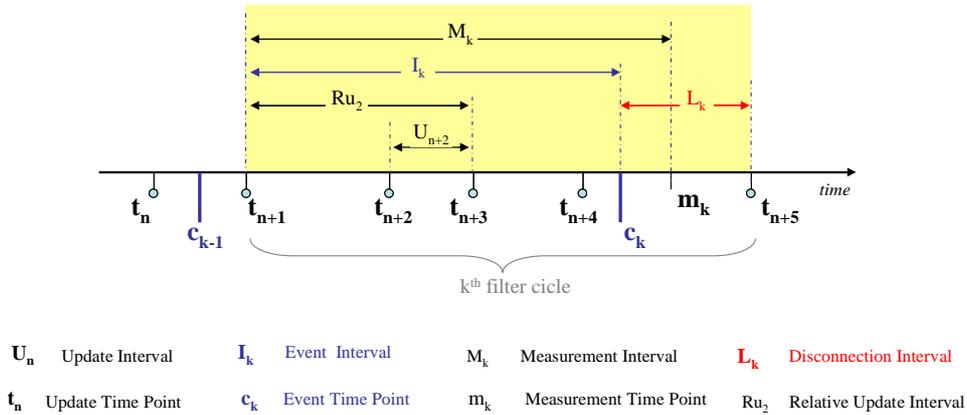


Figure 7.2: Abstraction

Definitions:

Beginning of the filter cycle is the Update Time point following the Event Time Point. In Figure 7.2, it is t_{n+1} and t_{n+5} .

Disconnection Interval (DI) is the interval between the Event Time Point (ETP) and the first following Update Time Point (UTP). The packets send by the mobile host in the DI get lost.

Update Interval (UI) is the interval between two following Update Time Points.

Event Interval (EI) is the interval between Event Time Point and the beginning of the filter cycle.

Absolute Event Interval (AEI) is the interval between two subsequent Event Time Points.

Relative Update Interval (RUI) is Update Interval relative to the beginning of the filter cycle.

Measured Time Point (MTP) is measured Event Time Point according the filter model. This is generated value depending on the algorithm described later in 8.4.1.2. It is called measured because it served as semi measurement point in the classical filter theory.

Measured Interval (MI) is interval between Measured Time Point and the beginning of the cycle.

Active Connection (AC) is ability to exchange packets bidirectional between the hosts, thus the PoAs are known to the both hosts. This is the opposite of *Disconnection Interval*, where the host cannot exchange packets.

Maximum Disconnection Interval (MDI) is the time without active connection tolerated by the application. This is typically a couple of seconds for real time applications, such as VoIP. For non real time applications, it can be some minutes, like email for example. The maximum Disconnection Interval value must not be exceeded. Even if there is less probability of disconnection, the maximum disconnection must not be exceeded. Practically, there must be an update at least once in this interval. Otherwise, there is a chance of disconnection with more then MDI. This value is set by the application. The interval is denoted as x_{\max} . is the mathematical equation for readability.

7.4 Targets

The Disconnection Interval (DI) is less than or equal to the smallest Update Interval (UI) surrounding the Event Time Point (Figure 7.2). It follows that the smaller the Update Interval the smaller the Disconnection Interval becomes. The target is to minimise the UI where the event (ETP) happens and to maximise the interval where it does not. This is done by making the update proportional to the PDF. The Maximum Disconnection Interval (MDI) must not be exceeded at the same time. The suggested methods consider also the hardware and network properties in parameters explained for every method in chapters 8, 9 and 10.

7.5 Simulation and performance evaluation

The simulation is proof of concept and shows the properties of the designed methods. It is done for the three methods using Matlab 7.0. The results are compared to updates at the constant intervals and in between. In this way, the qualities of the new method are verified.

Fair comparison is achieved using the same resources for the methods, thus the same number of updates during the time of simulation. Every method runs with the same number of updates in the same experiment time. The better the method is, the shorter the Disconnection Interval. Statistics of the disconnection intervals are compared, such as mean, max, min, variance etc.

Analysis of the results is provided in 8.6, 9.5, 10.9 and 12.3. There is no best performance or winner algorithm (see 12.3) since the optimal algorithm varies for the different usage scenarios. The algorithms choice depends on the concrete deployments. It is also possible to switch between the algorithms.

7.5.1 Simulation input data

Many simulations were carried out to test the properties of the algorithm. Five of them were chosen to demonstrate the qualities of the new algorithms. They are used to test all suggested algorithms, see chapters 8, 9 and 10. In this way, the results of the algorithms can be compared to each other. The first four input data (movements) are generated with mathematical equation. The last case is simulated with real dialup access statistics.

Representative scenarios are difficult to choose since the scenarios are consequence on the usage. The M-VPN can be deployed, for example in cellular networks in 3GPP, mobile vehicle with WLAN or mobile smart devices in campus building (office). The speed of IP/port changes and the moving patterns are different. Furthermore, the M-VPN can be used in VoIP environment with requirement of minimal disconnection or non real-time application like HTTP information with low requirement on disconnection. The chosen four simulation cases represent the most common cases building the primitive scenarios, which are:

- 1) In the first scenario, the intervals between the ETPs are white noise with some constant shift. Figure 7.3 presents a practical case where a mobile host is moving with a constant speed in a wireless environment, like a car moving at direct road. The wireless access networks cover similar areas transparently connected to the Internet. The changes of the PoA are at semi equal intervals because the wireless areas and the host speed cannot be constant in the praxis. For this reason, there is a deviation expressed in white noise. The white noise represents the differences between the PoA changes. No doubt, there could be other representations of this case. The equations are expressed by:

$$aEI_r = |N(200, \sigma)| ,$$

$$\sigma = 10$$

where aEI_r is the r^{th} Event Interval. $N()$ denotes normal distributed random values with a mean value of 200 sec and a standard deviation of 10 sec. The mean of 200 means the host stays mean 200 sec in wireless area. Practical representation of these values is for example: wireless cell (aria) with radius less has 1,4 km and the car (mobile host) moving with 50 km/h. The deviation of 10 gives changing intervals mostly from 170 sec to 230 sec, which represents the variation of the speed and size of the wireless cell.

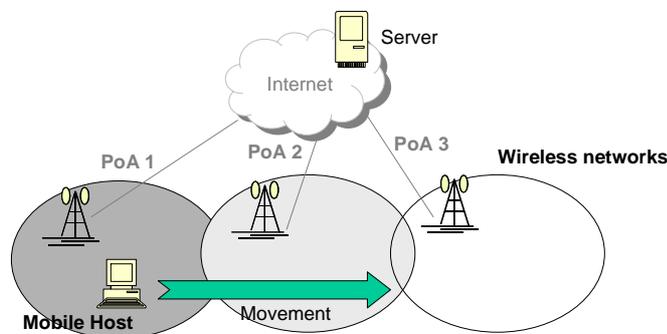


Figure 7.3: Host moving in wireless environment with constant speed

- 2) In the second scenario, the intervals of PoA change (ETPs) are generated by two rotating white noise sources. A practical representation could be similar to the

previous case with an additional NAPT as an intermediate devices. The wireless access points typically have an NAPT device, which can lead to PoA change before the host leaves the area. The PoA can be changed due to host movement or an NAPT table reset. This is shown in Figure 7.4. There are two white noise sources, which are rotating. They can be described by:

$$aEI_r = |N(100, \sigma_1)| \text{ or } |N(100, \sigma_2)|$$

$$\sigma_1 = 10, \sigma_2 = 10$$

The operator *or* denotes the rotation with 50% probability. Half of the values are generated by the first distribution and half by the second. The mean values of 100 sec gives the mean of PoA change. A practical representation may be a wireless area (cell) with radius of 0,7 km and car (mobile host) speed of 50 km/h. The deviation of 10 represents the variation of cell size and speed. The values are the values mostly between 70 sec and 130 sec.

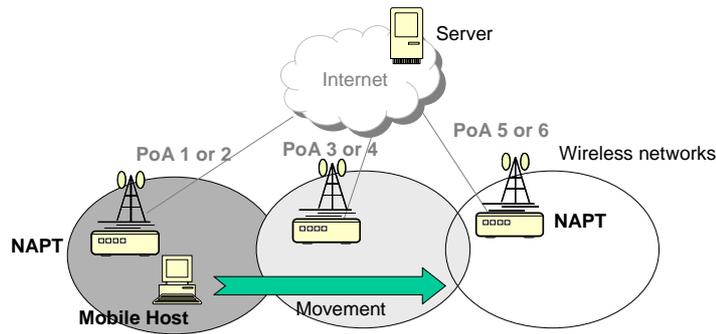


Figure 7.4: Host moving in wireless environment with NAPT

- 3) In the third scenario, sinus based Event Intervals with some white noise are defined. This occurs by repeating patterns in the Event Intervals, like a day cycle. The intervals are small in morning hours and large at night. The values go smoothly between these states. A sinus-based signal with white noise is defined as:

$$aEI_k = \sigma \sin\left(4\pi \frac{k}{S}\right) + \frac{7}{10} \sigma \sin\left(\frac{3}{2} \pi \frac{k}{S}\right) + \sin c\left(\frac{k}{S}\right) + N\left(500, \frac{\sigma}{2}\right)$$

The variable S indicates the total number of samples, $S=5000$. The number of samples is set to 5000 because it is sufficiently large to generate meaningful results, thus the results do not change significantly with increasing the number of samples. The k is the index of the calculated sample. The σ is equal to 10 to and represents minimal changes of the day cycle.

- 4) In the fourth scenario, recursive non-linear values with added white noise are the intervals between the ETPs. This is very challenging for every prediction method. It can represent a complex movement of the host with out any evident simple logic, shown in Figure 7.5. (The equation is also used in [6]). The equation defined as:

$$aEI_k = \frac{aEI_{k-1}}{2} + \frac{25r_{k-1}}{1+r_{k-1}^2} + 8\cos(1.2k) + N(200,20)$$

- 5) The fifth experiment is made with real data gathered from L2TP over IPsec dial up network, thus remote access. The intervals between the ETPs are the user's remote login in the corporate network. The deployment M-VPN can be very different from cell phone to trains. The remote access expresses the relocation of the mobile phone to some degree. There is a peak in the morning and afternoon. In the night, there is

less activity. The anonymous data was available to the author and used for the testing.

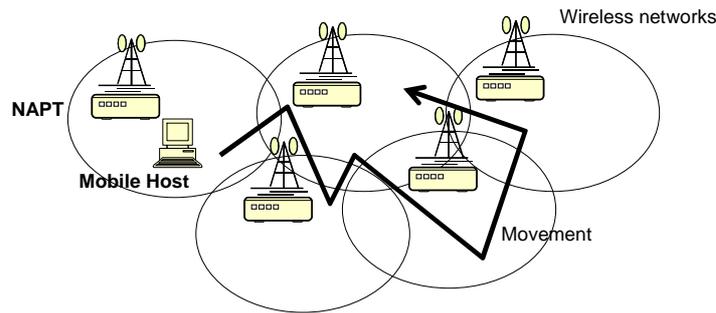


Figure 7.5: Complex mobile host movement

The constant parameters in the simulation are chosen to represent real cases, where small disconnection is required and the mobile host moves fast. For the first four experiments, the following parameters are used: The application requires 5 sec of Maximum Disconnection Interval (MDI). The simulation is made with 5000 samples, thus there were 5000 prediction cycles. The specific parameter are defined in 8.6, 9.5 and 10.9

7.5.2 Statistics and diagrams by the simulation

There are four diagrams showing the properties of the simulated algorithm. They are presented for each simulation case in 8.6, 9.5 and 10.9. The first diagram shows the Event Intervals and the posterior estimated Event Intervals (see example in Figure 7.6). It can be used for the optical evaluation of signal and the prediction.

A second diagram shows the histogram of the Disconnection Intervals (DIs) and gives qualitative numerical results (see example in Figure 7.7). This is the most important diagram since it compares the current method to the constant Update Intervals in a fair way. The fair way means that the same resources are used in the suggested method and in method with constant intervals. Both methods use the same number of location updates for the same time of simulation. In this way, the DIs can be directly compared between the methods which allows qualification of the better algorithm.

The constant update algorithm is the simplest and mostly used method, see chapter 1. The nodes send updates in constant intervals, thus there is no optimisation. The maximum Disconnection Intervals (DI) is equal to the Update Interval. This is reference method and all suggested methods are compared to it.

The DI corresponds to an error in the filter theory and must be minimised. The distribution of the DI is presented in the histogram on the second diagram. The following vertical lines mark the following values:

- A dotted black line shows the mean Update Interval (UI), thus the mean of maximal disconnection.
- A dashed black line shows the maximal Disconnection Intervals (DIs) by constant Update Interval see Variable *const UI*.
- A red dashed line marks the maximum Disconnection Interval (error) in this simulation by the method.
- A grey dotted line shows the user defined maximum disconnection (MDI).

Certain very important qualitative values are shown in the bottom-right corner of the histogram. Their definitions are:

Mean Disconnection Interval {mean DI} is the mean of all Disconnection Intervals, see 2.2. The mean DI must be minimised and it corresponds to the error in filter theory. The value can be calculated only in simulation environments since the Event Time Point is unknown to the hosts in practical environments.

Mean Update Interval {mean UI} is the mean of the Update Intervals containing the Event Time Point. The value is measurable at the hosts and can be used in real environments. Minimising the mean UI consequently minimises the mean DI.

Maximum Update Interval {max UI} is the maximum of all Update Intervals during the simulation.

Maximum user defined Disconnection Interval {max user def DI }.The maximum disconnection tolerated by the user application, denoted as x_{\max} .

Maximal disconnection by constant Update Interval {const UI} is the maximum disconnection achieved by constant update method using the same resources as the suggested method. The value is calculated as follows: Let us assume that t_{meas} is the total simulation time, thus the sum of all generated intervals (EIs). The number of updates during the simulation is denoted as N_{LUs} . The updates are uniformly distributed at constant intervals. The Update Interval is t_{meas}/N_{LUs} . It is the maximal Disconnection Interval by constant UI with the same resources.

Mean disconnection by constant Update Interval {const UI mean DI }. This is the mean of DI by constant UI with the same resources.

Proportion of mean Update Intervals by constant updates and the mean Update Interval by suggested method {const UI / mean UI}. This value gives the ratio between the const UI and mean UI. This is a direct comparison of the performance of the algorithm in a simple way. The higher the value the better the algorithm.

Update Intervals with ETP smaller than constant Update Intervals {LU < const UI}. This is the number of UI with ETP smaller than the maximum disconnection by constant interval (const UI). The value is a percentage of the total number of ETPs. This is a simple direct comparison of how good the filter is to constant LU interval. If the value is fifty percent, then there is probably no difference between the methods. This value gives a very raw comparison.

The DI for filter by sample is plotted at the third diagram, see example in Figure 7.8. The last forth diagram shows the PDF constructed by the suggested method and the PDF constructed by the Event Time Points (Figure 7.9). Both functions must be proportional. The second PDF is normalises to the first one, so that the maximum values fits. In this way, the curve forms can be compared. The more the functions look similar, the better the quality of the update is.

7.6 Porting of updates algorithm in further protocols

The location update protocol creates a new framework, which can be deployed in many other protocols. The update problem and its solution with M-LU are not specific to mobile

environments. The M-LU contributes to all procedures, which can be abstracted according to 7.3. An example of protocols with potential M-LU deployments is shown in chapter 11. These can be routing protocols like BGPv4 or DPD in IPSec, or even Mobile IP.

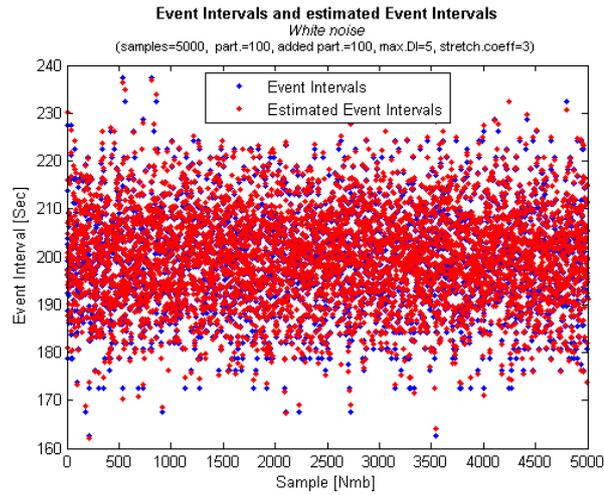


Figure 7.6: Example of Event Intervals and estimated Event Intervals

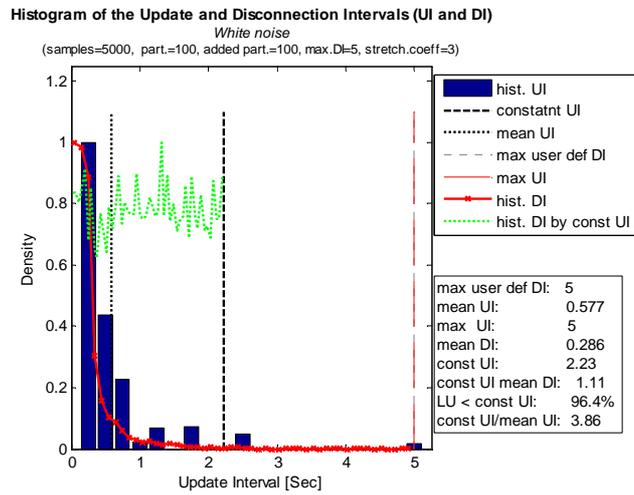


Figure 7.7: Example of result histogram

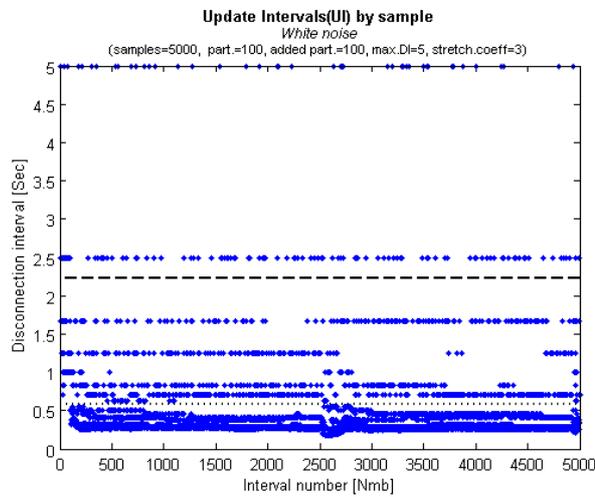
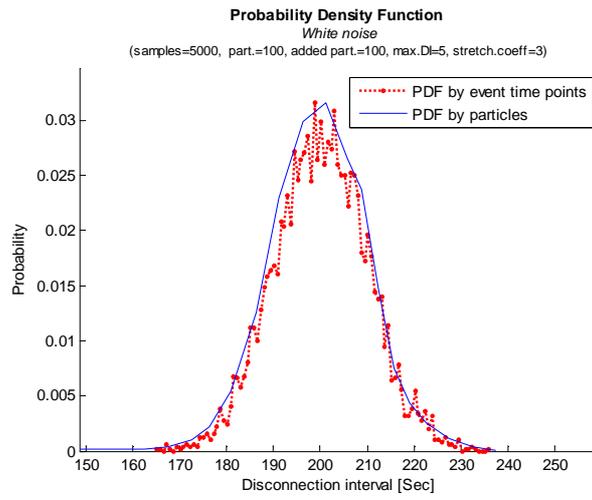


Figure 7.8: Example of UIs



7.7 References in chapter 7

- [1] Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems", ASME, 1960
- [2] Gelb, A., "Applied Optimal Estimation", MIT Press, 1974
- [3] Wiener, N., "The Extrapolation, Interpolation and Smoothing of Stationary Time Series," John Wiley & Sons, Inc., New York, N.Y., 1949.
- [4] Mouzouris, G., "Designing fuzzy logic systems," invited paper, IEEE Trans. on Circuits and Systems, Part II, vol. 44, pp. 885-895, Nov. 1997.
- [5] Maskell, S., Gordon, N., et al, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking", IEEE, VOL. 50, NO. 2, 2002
- [6] Forsyth, D.A. and Ponce, J., "Tracking with non-linear dynamic models", draft version of "Computer Vision - A Modern Approach", Prentice-Hall, 2002.

8 Mobile Location Update with Sequential Monte Carlo methods

This chapter presents a framework for setting the Update Time Points proportional to the Probability Density Function (PDF) of PoA change as defined in chapter 7. The PDF is estimated using the Sequential Monte Carlo (SMC) method also known as Particle filter or Sequence Importance Resampling (SIR).

The section 8.2 provides an overview of the Sequential Monte Carlo method and in particular the Sequence Importance Sampling. The implementation of Particle filter in M-LU is not straightforward, as already discussed in chapter 7. The new method for M-LU is defined in 8.4. The qualities of the new method are verified in a simulation described in 8.5.1. The results of the simulation are presented in 8.6.

8.1 Contributions

The author's contributions can be found in many parts of this chapter. The major theoretical ones are: First, the development of a model for an implementation of Particle filter in mobile updates, see 8.4.1. Second, the calculation of the update time-points dependent on the PDF of EI, see 8.4.2. Third, the new algorithm for update of the weights is implemented in 8.4.4. Forth, new mechanism for moving the particles is introduced in 8.4.5. A practical contribution is the simulation of the method and comparing the results to the constant updates in 8.5. The author has published the method in IEEE and part of the simulation results in IEEE [15].

8.2 Theory of Particle filter

8.2.1 Bayesian estimation

The estimation and prediction of values is an important part of the real world. Unfortunately, the knowledge for the most natural phenomenon is limited. It is practically impossible to consider all parameters of influences, like for example in the weather forecast. Due to these limitations, the models are simplified in general forms with some random variables representing the parameters, which are not directly considered. Models developed for a phenomenon allow the formulation of the Bayesian estimation.

8.2.2 Bayesian inference

The relationship between the conditional probability and the marginal probability distributions is provided by the famous Bayesian theorem. It can be interpreted as an equation for recalculating the belief in a certain hypothesis when obtaining new evidence. This interpretation is a core idea in the creation of inference process for hypothesis probability using evidence.

Bayesian inference uses the degree of belief in an event before (prior) the evidence has occurred and calculates an estimation of belief in the event after the evidence (posterior) has occurred. In other words, the Bayesian inference provides the relationship between the prior and the posterior distribution using likelihood characteristics. Mathematically it is defined as:

$$p(H_0 | E) = \frac{p(E | H_0)p(H_0)}{p(E)},$$

where H_0 is the prior or null hypothesis and E is the evidence.

$p(H_0 | E)$ is the posterior conditional probability of H_0 given E .

$p(E|H_0)$ is the likelihood function, the conditional probability of E given H_0 .

$p(H_0)$ is the prior probability of the hypothesis.

$p(E)$ is the marginal probability of E , i.e. the probability of E under all mutually exclusive hypothesis.

The evidence is the observed value, mostly called measurement in the filter theory. It is denoted as y further in this text. The hypothesis is the target value, thus the value of interest. It is an unobservable value, referred to x in this text. The relation between observation and unobservable value manifests figurative in FM radio example: The antenna measures values of signal and noise, thus the evidence y . We are interested only in radio signal x without the noise, called hypothesis. Only the signal cannot be measured directly and must be extracted from the antennas output.

In our case, the observed values are not available continuous in the time. The values are obtained sequential on certain intervals. Here, y_k denotes the k^{th} observation. The sequence of observations 1 to k is denoted as $y_{1:k}$. Following the same logic, the unobserved values become also available sequential and are referred as $x_{1:k}$. The behaviour model for unobserved values is called Hidden Markov Model (HMM).

Every time a new measurement y_k is received, a recursive filter is deployed to calculate an estimate of the target signal x_k . The advantage of the recursive calculation is that it is not necessary to store all past measurements. It is not necessary to recalculate from scratch. The result is updated with the new value.

The filter consists of two stages: prediction and update. The prediction stage uses the system model to predict the future state. The word prediction is synonym for prior estimation in this thesis. In the update phase, the PDF of signal is recalculated considering the received measurement, thus posterior estimation or update. Both prediction and update phases are executed sequentially.

In the prediction phase, it is assumed that posterior the PDF function at time $k-1$ is known $p(x_{k-1}|y_{1:k-1})$. The Chapman-Kolmogorov equation gives the prior estimation (prediction):

$$p(x_k|y_{1:k-1}) = \int_{-\infty}^{\infty} p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1},$$

After a measurement y_k becomes available, an update for the posterior estimation can be done using Bayesian rule for calculation the posterior density:

$$p(x_k|y_{1:k}) = \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})}, \text{ where}$$

$$p(y_k|y_{1:k-1}) = \int_{-\infty}^{\infty} p(y_k|x_k)p(x_k|y_{1:k-1})dx_k$$

This is the optimal solution in all aspects, but has only a theoretical and general character. The integral cannot be solved for all density functions and therefore, it cannot be exactly calculated.

8.2.3 Problem statement

The Bayesian inference cannot be solved analytically for all possible cases, because it requires the evaluation of complex high-dimensional integrals. For particular cases, such as a

linear model with Gaussian noise, an analytical solution has been available since the end of the 50s - the famous Kalman filter [1]. The solution is optimal in all aspects because it uses the optimal estimation known for linear systems. Unfortunately, most natural processes are non-linear and without Gaussian noise.

Many suboptimal solutions were proposed considering non-linear systems. The Extended Kalman Filter (EKF) is a typical example described in chapter 10. The Extended Kalman Filter approximates the non-linear equations to linear ones using truncated Taylor series. Through this approximation, an analytical solution was found to the problem. The filter is called suboptimal since its efficiency strongly depends on the approximation of the signal. The algorithm suffers from a lack of adequacy in some cases as shown in [2].

8.3 Monte Carlo methods

Monte Carlo methods solve the Bayesian inference in a different way. With the increase in the computational power of the computer, the Monte Carlo Methods become a reasonable alternative to the analytical ones.

The source idea is that the PDF of every function can be calculated from a sufficient number of samples (observations). This property is used in the Monte Carlo integration method. The PDF of the PoA changes is estimated in a Location Update context.

The clear advantage is that the method performs for non Gaussian and non-linear models. A disadvantage is that the convergence of the method depends on the number of observations and the choice of importance function as shown in 8.3.3.

8.3.1 Sequence Importance Sampling (SIS)

The Sequence Importance Sampling (SIS) is the most basic of all sequence Monte Carlo filters implemented nowadays. Various names are used to refer to this method, such as Bootstrap filter, condensation algorithm or Particle filtering. In our text the term Particle filter is used, since it is the most popular one.

The Particle filter is a recursive Bayesian approach with Sequential Monte Carlo (SMC) method. The key idea is to represent the required probability density function by a set of random samples with associated weights. In this way, the SIS filter approaches the optimal Bayesian estimate.

Lets assume that N independent measurements $\{x^i\}_{i=1}^N$ are available for an observed phenomenon. The posterior density of the samples can be expressed by:

$$p(x)_N = \frac{1}{N} \sum_{i=1}^N \delta(x - x^i)$$

Here, $\delta_{(x-x^i)}$ is the delta-Dirac mass located at x^i . Regarding the strong law of the large numbers, the calculated discrete distribution converges to the signal density distribution with a sufficiently large number of measurements.

$$p(x)_N \xrightarrow[N \rightarrow \infty]{} p(x)$$

The number of observations to achieve almost certain convergence is extremely large. In practice, an algorithm with fast convergence is needed.

The observation can be presented by $\{x^i, w^i\}_{i=1}^N$ where w^i denotes the weights of the samples x^i . The weights stress the high values of the PDF. The weights are normalised, thus $\sum w^i = 1$. The PDF can be expressed as:

$$p(x) \xrightarrow{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N w^i \delta(x - x^i) \quad \text{Equation 8.1}$$

The choice of the weight coefficients is made according to the importance sampling [9]. The weights are split into two components.

$$w^i = \frac{w^{*i}}{\sum w^{*i}}, \quad w^{*i} = \frac{\pi(x)}{q(x)}$$

Here, $\pi(x)$ is proportional to the distribution of $p(x|y)$, thus $\pi(x) \propto p(x|y)$. The sign \propto means proportional. The $q(x)$ has the distribution of x , thus $q(x) \sim x$. The sign \sim means same distribution. The $q(x)$ is called importance function.

This representation is the base for particle definition. Each particle consists of one weight coefficient and multiple unobserved values (hypothesis). The particles are independent and together build the PDF. The weight stresses the importance of this particle, i.e. if the PDF has a high value at this point. The measurements (observations) become available sequentially in periods. A new value x is added in particle each time a measurement y becomes available. By receiving a new value, the particle weight must be recalculated. If there are 0 to k observations, there will be $k+1$ values in every particle, namely $\{x_{0:k}^i, w_k^i\}_{i=1}^N$. The Equation 8.1 can be rewritten:

$$p(x_k | y_k) \xrightarrow{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N w_k^i \delta(x - x_{0:k}^i) \quad \text{Equation 8.2}$$

$$w_k^{*i} = \frac{\pi(x_{0:k}^i)}{q(x_{0:k}^i)} = \frac{p(x_{0:k}^i | y_{1:k})}{q(x_{0:k}^i | y_{1:k})}$$

The data is obtained sequentially and the filter coefficients must be recomputed every time data becomes available. Increasing the number of observations increases the required resources to store all past values. After a certain point, they are not affordable even for the modern computers. To solve this issue, the important density function must be factorised into recursive multiplication. In this way, the following values are calculated recursively from the previous one.

$$q(x_{0:k} | y_{1:k}) = q(x_k | x_{0:k-1}, y_{1:k}) \cdot q(x_{0:k-1} | y_{1:k-1}) \quad \text{Equation 8.3}$$

The equation gives the possibility for factorised calculation of $x_{0:k}^i \sim q(x_{0:k} | y_{1:k})$ based on the previous values $x_{0:k-1}^i \sim q(x_{0:k-1} | y_{0:k-1})$.

$$q(x_{0:k} | y_{1:k}) = q(x_k | x_{0:k-1}, y_{1:k}) \prod_{t=1}^{k-1} q(x_{0:t} | y_{1:t}) \quad \text{Equation 8.4}$$

The update of the importance weight is achieved using the Bayesian equation introduced shortly in 8.2.2:

$$p(x_k | y_{1:k}) = \frac{p(y_k | x_k) p(x_k | y_{1:k-1})}{p(y_k | y_{1:k-1})}$$

$$p(x_{0:k} | y_{1:k}) = \frac{p(y_k | x_k) p(x_k | x_{0:k-1}) p(x_{0:k-1} | y_{1:k-1})}{p(y_k | y_{1:k-1})}$$

$$p(x_k | y_{1:k}) \propto p(y_k | x_k) p(x_k | x_{k-1}) p(x_{0:k-1} | y_{1:k-1}) \quad \text{Equation 8.5}$$

By substituting Equation 8.5 and Equation 8.3 in Equation 8.2 the update for the weighted coefficients is obtained:

$$w_k^{*i} \propto \frac{p(x_{0:k}^i | y_{1:k})}{q(x_{0:k}^i | y_{1:k})} = \frac{p(y_k | x_{0:k}^i) p(x_{0:k}^i | x_{0:k-1}^i) p(x_{0:k-1}^i | y_{1:k-1})}{q(x_k^i | x_{0:k-1}^i, y_{1:k}) q(x_{0:k-1}^i | y_{1:k-1})}$$

$$w_k^{*i} \propto w_{k-1}^{*i} \frac{p(y_k | x_{0:k}^i) p(x_{0:k}^i | x_{0:k-1}^i)}{q(x_k^i | x_{0:k-1}^i, y_{1:k})} \quad \text{Equation 8.6}$$

8.3.2 Degeneracy of SIS

The main drawback of the SIS is that the variance of the weights cannot decrease. The variance increases or stays constant over the time. After a few interactions, all coefficients, except one, become zero in practical terms. The computational power is wasted in calculating particles having zero weights and therefore are useless in the PDF calculation. The degeneracy problem cannot be avoided and will persist for all $q(x_k^i | x_{0:k-1}^i, y_{1:k})$ functions to some degree.

This is a property of the importance function defined with recursion in Equation 8.3. The increase of the variance of the importance function leads to an increase in the variance weight coefficients, when the weights are interpreted as a random variables. This is obtained directly from the definition:

$$\text{var}(w_k^{*i}) = \text{var}\left(\frac{p(x_{0:k}^i | y_{1:k})}{q(x_{0:k}^i | y_{1:k})}\right)$$

When the variance of $q(x_{0:k}^i | y_{1:k})$ grows, then the variance of division also grows and the variance of the weights grows consequently. Increasing variance means the difference between the values increase. When the difference increases and the weights are normalised, some values go to zero and some values to one. There are no intermediate values in the interval.

The effective sample number was defined to measure the level of degeneracy size [9, 8]:

$$N_{eff} = \frac{N}{1 + \text{var}(w_k^i)}$$

The variance of the weights can be estimated:

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^N (w_k^{*i})^2} \quad \text{Equation 8.7}$$

The effective size is always smaller than the number of particles N . The degeneracy problem increases when the effective sample size decreases. The defined equation is very important because it quantifies the degeneracy problem.

Observing the definition at Equation 8.7, it can be concluded that increasing the number of particles automatically decreases the degeneracy. This property cannot be used to stop the degeneracy directly, but urges use of a large number of particles.

The effect of the degeneracy can be limited trough: (1) a good choice of the importance function or (2) resampling. Both of them are described in the next chapters.

8.3.3 Choice of the importance function

There is no restriction for the importance function: every function leads to convergence. The question is to find a function, which performs well for a small number of samples and particles. Some of the functions lead to convergence for a very large number of samples. These functions cannot be practically used.

In order to limit the degeneracy, the common strategy is to minimise the variance growth of the importance function. Zeroing the variance of importance function stops the degeneracy of the weights. This strategy provides the best results because it adopts the importance function to the posterior PDF. The variance of the weights regarding the importance function is [4]:

$$\text{var}_{q(x_k|y_{1:k}, x_{k-1}^i)}(w_k^i) = (w_k^i)^2 \left[\int_{-\infty}^{\infty} \frac{(p(y_k|x_k)p(x_k|x_{k-1}^i))^2}{q(x_k|y_{1:k}, x_{k-1}^i)} dx_k - p^2(y_k|x_{k-1}^i) \right]$$

The variance becomes zero for $q(x_k^i|y_{0:k-1}, x_{1:k}^i) = p(x_k^i|y_k, x_{k-1}^i)$ as shown in [4]. This is the optimal importance function. The Equation 8.6 with the new importance function can be rewritten considering that the weight is dependent only on the last particle value k and not all values 0 to k gives:

$$w_k^{*i} = w_{k-1}^{*i} \frac{p(y_k|x_k^i)p(x_k^i|x_{k-1}^i)}{p(x_k^i|x_{k-1}^i, y_k)} = w_{k-1}^{*i} p(y_k|x_k^i)p(x_k^i|x_{k-1}^i) \left(\frac{p(y_k|x_k^i, x_{k-1}^i)p(x_k^i|x_{k-1}^i)}{p(y_k|x_{k-1}^i)} \right)^{-1}$$

$$w_k^{*i} = w_{k-1}^{*i} p(y_k|x_{k-1}^i)$$

The optimal function has some limitations. The first shortcoming is that we need to sample from the $p(x_k^i|x_{k-1}^i, y_k)$ to calculate the PDF function. The second problem is the calculation of the $p(y_k|x_{k-1}^i)$:

$$p(y_k|x_{k-1}^i) = \int_{-\infty}^{\infty} p(y_k|x_k)p(x_k|x_{k-1}^i)dx_k$$

This integral cannot be solved analytically for all functions. A solution can be found in some particular cases. When x_k has finite states, then the integral becomes the sum through all states. The integral and the $p(x_k^i|x_{k-1}^i, y_k)$ can be solved numerically. In general, if the model is with finite states, then the optimal importance function can be used.

There is an existing solution of the integral when observation equations are a linear and Gaussian noise mixture. The integral $p(y_k|x_{k-1}^i)$ can be analytically solved. The values $p(x_k^i|x_{k-1}^i, y_k)$ can be calculated. Notice that there are no restrictions for the transition equation (hidden Markov equation).

Implementation of the optimal importance function gives the best results and limits the degeneracy problem. Unfortunately, it is not possible to use the function in M-LU, because the model is neither linear nor with finite states which are the requirements for optimal solution.

8.3.3.1 Suboptimal importance functions

The optimal importance function cannot be used in non-linear models with infinite states as in M-LU case (see 8.3.3). Suboptimal functions and methods are implemented for these scenarios.

One suggested method is with approximation to linearisation. This is called the local linearisation method. It works in a very similar way to the Extended Kalman Filter. The non-linear observation equation is approximated with truncated Taylor series.

A different approach is the choice of a convenient importance function. The most popular one is using the prior density function [3, 4]:

$$q(x_k^i | x_{k-1}^i, y_{1:k}) = p(x_k^i | x_{k-1}^i)$$

$$w_k^{*i} = w_{k-1}^{*i} p(y_k | x_k^i)$$

Using the prior density seems a reasonable alternative because it is easy to implement. It is used in this thesis for the M-LU method with Particle filter.

There are other alternative importance functions and methods suggested for different non-linear methods. There is not an universal function covering all cases. The right choice of method and importance function is critical for the performance of the algorithm.

8.3.4 Resampling

The degeneracy becomes a significant problem for the implementations when using suboptimal importance functions. Resampling is method for solving the problem by suboptimal function. The idea of resampling is to eliminate particles with small weights values and to stress ones with high weights. The resampling involves generation of a new set $\{x_k^{*i}\}_{i=1}^N$ where x_k^{*i} denotes new values. The new set approximates the posterior PDF given by:

$$p(x_k | y_k) \approx \frac{1}{N} \sum_{i=1}^N w_k^i \delta(x - x_k^i)$$

The new set approximates the PDF function without certain particles. Generally, it is discrete density sampling:

$$p(x_k | y_k) \approx \frac{1}{N_r} \sum_{i=1}^{N_r} \delta(x - x_k^{*i})$$

The number of new particles - N_r can be different after the resampling. To keep it simple, we assume $N_r = N$. To get a parallel to $\{x^i, w_k^i\}_{i=1}^N$ the new generated sets also have weights which are all equal to $1/N$. After the resampling, the particles have the form $\{x^{*i}, 1/N\}_{i=1}^N$. The weights coefficients are reset, so the degeneracy problem disappears.

The resampling from the PDF can be carried out in different ways. The straightforward method is taking N samples uniformly distributed in $[0, 1]$, denoted as $\{r_i\}_{i=1}^N$. For every sample point r_k , where $k=1$ to N , the new particle values are selected by:

$$\text{if} \left(\sum_{i=1}^{l-1} w_k^i < r_k < \sum_{i=1}^l w_k^i \right) : x_k^{*k} := x_k^l$$

The Figure 8.1 graphically shows the algorithm. The intervals at the abscise (x-axis) are equal to the weight-coefficients, so they build a sector representing the i^{th} particle. The sum of the weights is one (1), thus they are normalised. The r is uniformly distributed, thus the

points have $1/N$ distance between them. The sector in which the r point is falling defines the value of particle. For example: in Figure 8.1, the second point of r is within the first sector. This means the new value of the second particle is equal to value of first current particle $x_k^{*2} = x_k^1$. The number of points falling in a sector gives the number of particles with the same value. Some old particle values will not be transferred in the new values. For example: in Figure 8.1, there is no r falling in the second sector, thus the second particle value will not be taken in a new particle.

Generally, particles with small weights are ignored and with large weights are emphasized. After resampling, the new approximated density converges with the original one.

This procedure solves the degeneracy problem but has theoretical and practical disadvantages. Theoretically: the particles are not independent variables as assumed in the beginning after resampling. This is disadvantageous for the diversity, since large weights are emphasized and small weights are discarded. Practically, the implementation of the filter cannot be parallelised. The parallelisation speeds up the processing. For resampling all particle and must be present.

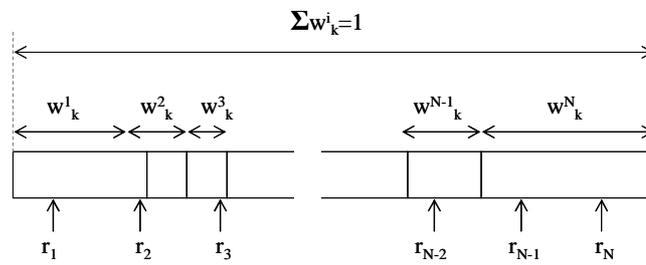


Figure 8.1: Resampling selection algorithm

Considering the mentioned disadvantages, it can be concluded that: resampling should be performed only when the degeneration becomes a significant problem. It could be done in all interactions, but this can lead to the tampering of the good results.

On which interaction a resampling should be carried out, can be chosen using the equation for degeneration quantification (Equation 8.7). A threshold value $N_{threshold}$ for the degeneration must be pre-defined. Resampling is done if the degeneracy gets higher than the defined threshold. This is expressed in the equation:

$$if \left(N_{threshold} < \frac{1}{\sum_{i=1}^N (w_k^{*i})^2} \right) : resample$$

8.3.5 Algorithm summary

The major steps by the Particle filtering are summarized at the following figure:

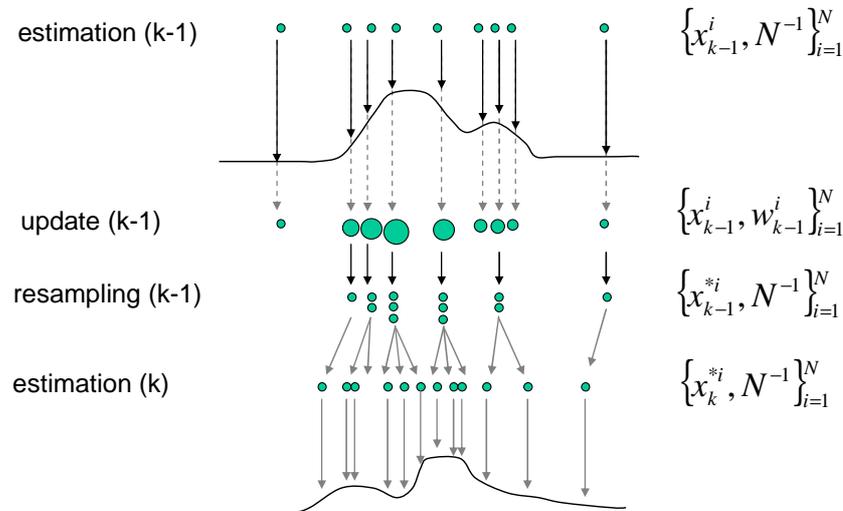


Figure 8.2: Particle filter

The number of particles N is defined before beginning the calculations. The number of particles gives the accuracy of the approximated function. On the one hand, a large number of particles decreases the degeneracy problem and improves the quality of the approximated function. On the other hand, the more particles are used, the more computational resources are required.

Distribution of N random samples $p(x_0)$ is generated for the first initialization. The weight coefficients are set to $1/N$. The N particles are $\{x_0^i, 1/N\}_{i=1}^N$ now. If there is no information about $p(x_0)$, any constant value can be used. This will slow down the convergence.

The update procedure is used to compute the weights after obtaining the first measurement. Most of the cases use sub optimal functions: $w_k^{*i} = w_{k-1}^{*i} p(y_k | x_k^i)$. After normalisation the results are set for the particle with weights: $\{x_k^i, w_k^i\}_{i=1}^N$. The PDF can be calculated using Equation 8.2. If the degeneracy size of the weights is higher than the threshold value, then resampling is performed.

The new state $k+1$ is predicted through the equation for the Hidden Markov Model (HMM). When the new measurement is obtained, the interaction is repeated from the beginning. The algorithm steps are written again in a semi program code:

Particle filter

- Initialisation

N samples with distribution $p(x_0)$ are generated.

The particles are:

$$\{x_0^i, 1/N\}_{i=1}^N$$

- Importance sampling

By receiving the measurements the weight are calculated:

$$w_k^{*i} = w_{k-1}^{*i} p(y_k | x_k^i),$$

where $w^i = \frac{w^{*i}}{\sum w^{*i}}$. The new set of particle is $\{x_k^i, w_k^i\}_{i=1}^N$.

The density function can be approximated.

- Resampling

$$\text{if} \left(N_{\text{threshold}} < \left(\sum_{i=1}^N (w_k^{*i})^2 \right)^{-1} \right):$$

For all N uniformly distributed $r[0,1]$

$$\text{if} \left(\sum_{i=1}^{l-1} w_k^i < r_k < \sum_{i=1}^l w_k^i \right) : x_k^{*k} := x_k^l$$

The result is $\{x_k^{*i}, w_k^i\}_{i=1}^N$

- Prediction

The new particles are estimated using HMM equation. For example :

$$x_{k+1}^i = f(x_k^i) + v \quad , \text{ where } v = N(0, \sigma^2)$$

Go to step *Importance sampling*

8.3.6 Rao-Blackwellisation

Particle filters suffer from degeneration as mentioned previously. To limit the degeneration either particles or resampling are commonly used. Both methods have some disadvantages. In this section, a further method is described, known as Rao-Blackwellisation based on the Rao-Blackwell theorem.

The Rao-Blackwell theorem describes that if we have an unbiased estimator, then it is always possible to find another unbiased estimator. The variance of this is lower by conditioning on a sufficient statistic. Proof can be found in [13].

Rao-Blackwell Theorem: *If random variables have the joint distribution $\pi(R, L)$, then for a function of interest $f()$:*

$$\text{Var}_{\pi}(\mathcal{E}_{\pi}(f(R)|L)) \leq \text{Var}_{\pi}(f(R))$$

The key idea of Rao-Blackwellisation [12] is to make use of the information inherent to the problem itself. The problem space is partitioned in two analytically inferring sub-spaces. Assuming partitioning the state-space of X into two sub-spaces $L(\text{leaf})$ and $R(\text{root})$, so that $x_k = (r_k, l_k)$. The chain rule of probability can be written:

$$p(x_k | y_k) = p(r_k, l_k | y_k) = p(r_k | l_k, y_k) p(l_k | y_k)$$

Let us assume that $p(r_k | l_k, y_k)$ can be updated analytically and therefore efficiently. The probability $p(l_k | y_k)$ can be calculated using Monte Carlo Sequence Methods, if the conditions for analytical update are not fulfilled. Partitioning decreases the variance.

For the update of $p(r_k | l_k, y_k)$, the L and Y must be Gaussian or have a finite state space for the existence of an analytical solution. The update can be carried out with an optimal method such as Kalman filter or jump Markov liners system [12]. The methods are optimal in minimising the Mean Square Error and cannot be out-performed by any other algorithm.

The graphic representation of dynamical Bayesian Networks in Figure 8.3 can be used to understand the Rao-Blackwellisation Particle Filter (RBPF). The Bayesian network is a directed acyclic graph where nodes represent variables and arcs represent dependence (relation) among the variables. Dynamic Bayesian Networks extend Bayesian Networks with the probabilities which evolve over the time. The transition equation of the state space can also be graphically represented.

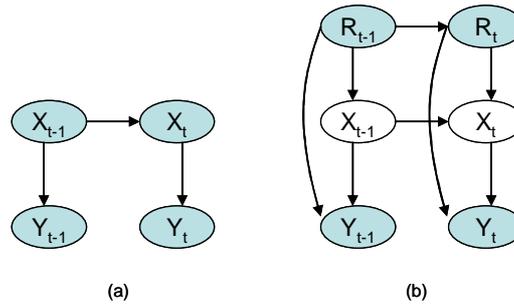


Figure 8.3: Dynamic Bayesian Network for Particle Filter with Rao-Blackwellisation

The simple Particle filter network with two time steps is shown in Figure 8.3 (a). The Particle Filter with Rao-Blackwellisation is represented in Figure 8.3 (b). The grey-blue cycles R are the variables, which we evaluate with the Particle Filters method. The white nodes are analytical updates.

The main initial difficulty with RBPF is how to identify the conditional relationship between the state variables, so that the state variables can be meaningfully partitioned into two groups. The second issue is what kind of analytical filter should be used to estimate the leaf variables conditional on the root variables. The RBPF can not be implemented in M-LU directly. It is described in this chapter for completeness of the topic.

8.4 Location Update procedure with Particle Filter

The implementation of the Location Update procedure is not straightforward for the Particle filter. There are some significant differences (see chapter 7), which are solved in order to use the mechanism.

8.4.1 Model

The natural phenomena are described with two models in order to be simulated. The first one describes the natural properties process, thus in our case the PoA changes. The second describes the measurement (observation). Typically, the observed variable is not the required one. The models for M-LU are described in the following chapters.

8.4.1.1 The natural properties of the Event Time Point

The natural properties of the PoA change can be expressed in this state-space equation. The equation describing the natural properties of the phenomenon is important to the efficiency of the algorithm.

The PoA changes are a reflection of the habits of the mobile user and properties of the access network. The PoA changes will usually follow the physical movements of the host. For example: The urban employee has typically high activity between 8:00h and 22:00h during the week. At night, the worker is less active.

In this simulation, no prior knowledge of the ETPs (PoA changes) is deployed. It is assumed that zero knowledge of the network and user's behaviour is present. This is the worst-case scenario and will show the qualities of the method well. The estimated particles are equal to the updated particle values.

8.4.1.2 Observation model

The second model describes the measurement (observation). Every measurement device has some characteristics, for example noise and dispersion. These characteristics are expressed in this model.

The measurement consists usually of the modified signal of interest and noise. In M-LU, the PoA changes (ETPs) cannot be measured, as described in 7.1. An observation model cannot be created directly. To solve this problem some assumptions are made, so that measurement values are produced. The measurement is created artificially to satisfy the classical methods and in the same time to represent reasonable values.

The distribution of the Event Time Point (ETP) within the Update Interval is unknown. The ETP can be everywhere in the Update Interval, see Figure 7.2. The only sure knowledge is that the ETP is within the interval with the event, but we require numerical value.

To work this problem out, we assume that the measurement value is in the middle of the Update Interval (LU). The maximum error in this case is the length of the Update Interval. A further important assumption is that the measurement error in the Update Interval (UI) is Normal distributed (Gaussian). The standard deviation is proportional to the length of the interval.

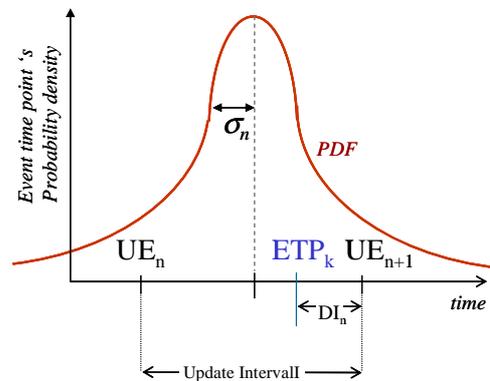


Figure 8.4: Measurement and PDF Function in update interval

The PoA change point and PDF are shown in Figure 8.4. The PDF function of the Gaussian error distribution is drawn in red, the famous Gaussian bell. It is important to stress that a PDF of M-LU distribution is infinite and it is not possible to be limited to an LU (update) interval. Practically, there is zero probability for the PoA occurrence outside the UI. This is a controversy since the Gaussian distribution is infinite and cannot be limited in finite interval. The Gaussian probability is used although it cannot satisfy this condition. There are not solid objective arguments but reasonable subjective ones. Big advantage of the Gaussian PDF is that it gives a smooth reduction of the probability. Values close to the LU interval are better weighted than values far from the LU interval, there is a smooth evaluation of the weight. Furthermore, many of the natural processes are Gaussian distributions.

The relationship between the size of the interval and the standard deviation is called stretching coefficient. The higher the stretching coefficient, the more values are considered outside the LU interval. Increasing the stretching coefficient decreases the degeneracy

problem, because it avoids strong particle concentration. Less degeneracy means less resampling. Less resampling is always good for the result, as described in 8.3.4. The disadvantage of a high stretching coefficient is the low PDF accuracy. A compromise in this coefficient must be found. There is no analytically optimal value. Depending on the application an interval between 1 and 3 times LU seems to give good results in our simulation.

The measurement is expressed using $f()$ function, which returns the middle of the Update Interval with PoA (ETP). The interval depends on the PDF of ETP and the ETP. Furthermore, the function returns the maximal measurement error as second value. The model equation is:

$$[y_k, e_k] = f(x_k, x_k^{1:N})$$

$$p(y_k | x_k) = N(f(x_k, x_k^{1:N}), c_{st} \cdot e_k),$$

where y_k is the measured value (in the middle of the Update Interval) and e_k denotes the maximal error (UI size). The first equation returns the measured values. The second equation expresses the probability distribution function. The $N(\mu, \sigma)$ is the Normal Gaussian distribution with mean μ and σ standard deviation. The standard deviation $c_{st} \cdot e_k$ depends on the maximum error multiplied by stretching coefficient c_{st} .

8.4.2 Location Updates distribution

The core idea is that the Update Intervals should be proportional to the probability of PoA change, as already discussed in chapter 7. In the areas, where probability is high, there should be high LU density. The principle is shown in Figure 8.5 for one filter cycle. The points must be proportional to the PDF of the Event Time Points (ETP).

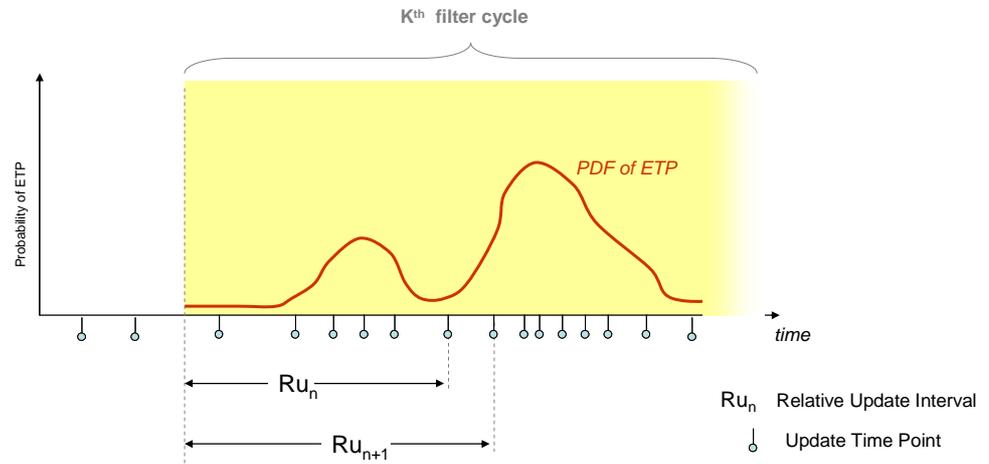


Figure 8.5: LU and PDF of ETP

The PDF built by the Relative Update Intervals Ru_i for M points, is:

$$p(u)_M = \frac{1}{M} \sum_{i=1}^M \delta(u - Ru_i)$$

The PDF of Event Intervals (EI) is defined by the Particle filter. The PDF for N particle is:

$$p(t)_N = \frac{1}{N} \sum_{i=1}^N w^i \delta(t - t_{par}^i),$$

where the t_{par}^i is the position of the particle.

The PDF by EI should be proportional to the PDF by the Relative Update Intervals (RUI) within the filter cycle. This is the ideal case, which can be expressed by:

$$p(t)_N = \frac{1}{N} \sum_{i=1}^N w^i \delta(t - t_{par}^i) \propto p(u)_M = \frac{1}{M} \sum_{i=1}^M \delta(u - Ru_i) \quad \text{Equation 8.8}$$

This is a key equation for determining the update points. The PDF functions must be only proportional. This is an important property used in chapter 8.4.3.

Observing Equation 8.8, the first suggestion is to set the Update Time Points (UTP) equal to the particle values (positions). Where the concentration of particle is high, there will be more updates, thus smaller UIs. It is absolutely correct and reasonable from a mathematical prospective. Unfortunately, it is not possible. The Update Intervals must not be greater than the Maximum Disconnection Interval (MDI) defined by the user, see 7.3. Direct copy of the values will not work, since there will be intervals bigger than the maximum disconnection. These are intervals with very low probability of ETP.

The condition that the Update Intervals must not be bigger than certain interval (MDI) is very critical. This means also that the PDF of the Relative UI (RUI) could not be proportional to the PDF by EI since there must be updates also in areas with zero probability of ETP. As a consequence, the condition of PDF proportionality is ideal. The RUI must be distributed almost proportional to the PDF of EI.

A practical suboptimal approach is suggested for defining the LU points, which is easy to implement. It delivers very good results although it is suboptimal. The simulation results are presented in 8.5.

The first step is to set an Update Time Point (UTP) at every $x_{bin} \leq x_{max}$ (smaller than Maximum Disconnection Interval). The condition for maximum disconnection is fulfilled in this way. The resulting intervals are called bins. A number of uniformly distributed UTP is added in each interval x_{bin} (bin) depending on the PDF constructed by the particle. If the PDF is high in the bin, then a large number of LU points is added. Certainly, the PDF is not a constant in the bin intervals. The reference value is the middle of the bin. Different reference values can be considered involving interpolation, mean etc. This is out of scope in this general method description.

The total number of additional UTPs added in all bins is defined by the user before the simulation begins. The higher the number added UTPs the lower the disconnection time is and therefore, the lower the measurement error is. Unfortunately, the higher the number of added updates the more resources are required. The performance of the algorithm becomes poor because of the wasted resources. If there are zero added UTPs, then there are constant intervals. It is the same as if no PDF considerations are done. Both extremes are bad for the performance. In current experience, values between 20 and 200 added LUs for 10 seconds of Maximum Disconnection Interval (MDI) produce good results.

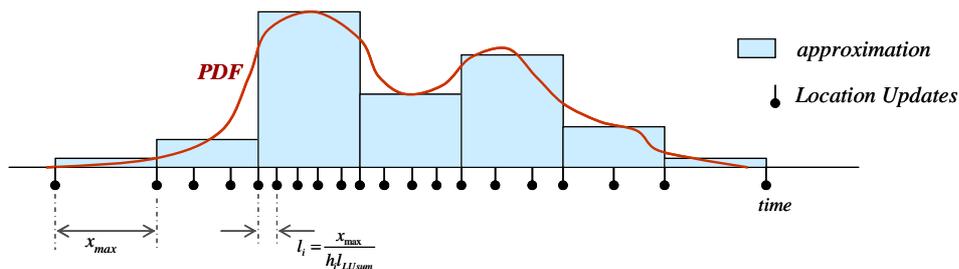


Figure 8.6: location updates distribution according the PDF

The total added UTPs is denoted as l_{LUsum} . The number added points in a bin depends on the PDF of the Particle filter as previously described. The PDF must be normalised before calculation. The number of added UTPs in the bin is calculated by multiplying the PDF value at the middle of the bin by the total number of added UTPs. The PDF value in the middle of the bins is denoted as $\{h_i\}_1^{bins}$. The number l_i of added updates in the i^{th} bin will be:

$$l_i = h_i \cdot l_{LUsum}$$

The distribution of the additional updates in the bin is uniform. There is no other information about the distribution, so uniform distribution seems to be the proper one.

The location updates and the PDF function are shown in Figure 8.6. The PDF in bin is approximated to constant values equal to the middle PDF values in the bin. The uniform distribution of the updates depends on bins high (PDF value in the middle). The size of the update intervals in the bin is division of x_{bin} to the number of added UTPs in bin:

$$u_i = \frac{x_{bin}}{h_i \cdot l_{LUsum}}$$

The UPTs can be calculated and executed in the following summarizing steps:

Location Update points

- PDF of the Event Time Point (ETP) with Particle filter is $\{x_k^i, w_k^i\}_{i=1}^N$
- At $x_{bin} \leq x_{max}$ intervals (bins) the UTP are set. These are the t_k^{Bin} , $k \in \aleph$
- PDF is approximated to the middle of the bins, h'_j , $j \in \aleph$
for (i=0 ; i<N , i++) {

$$h'_i = pdf\left[\frac{(t_i^{Bin} + t_{i+1}^{Bin})}{2}\right]$$
- Normalisation
for (i=0 ; i<N , i++) {

$$h_i = \frac{h'_i}{\sum h'_i}$$
- Number of added UTPS in the bin j is:

$$u_i = \frac{x_{bin}}{h_j \cdot l_{LUsum}}$$
- Location Update position t_i^{add} in bin j is
for (i=0; i<l_j; i++) {

$$t_i^{add} = j \cdot x_{bin} + i \cdot u_j$$

The size of the bin in the time is very important for the algorithm's performance. It can be equal to the x_{\max} . In this case, the size is influenced by the user and not by the algorithm itself. This can lead to underperformance when the bins are larger than the Event Interval. This can be easily proven in the extreme case, where there is only one bin. On the other side if the value is smaller than the x_{\max} , there is underperformance. The reason is that there are UTPs at x_{bin} also in areas where the probability of ETP is zero. In this areas, the update must be as rare as possible, thus best case at x_{\max} .

In our simulation we set $x_{bin} = x_{\max}$. The optimal x_{bin} depends on the movements and therefore on the usage. For example: a VoIP application over M-VPN in car with WiFi connection.

8.4.3 Resampling and particle position degeneration

The resampling limits the degeneracy problem through reordering the particle values in areas in which the PDF function is high. The weight coefficients are reset to constant and the degeneracy is eliminated. The PDF is constructed only by the distribution of the particle values and the weight plays no role at this stage.

Resampling is not suitable for the Location Update procedure. In case of zero knowledge of the natural model, the degeneration problem moves from the weight coefficients to particle position values. The variance of the particle decreases.

After few interactions the particle concentrates on a few values, thus the majority of the values are the same. The main reason is the missing noise component in the estimation prediction model. For comparison in literature, there is always noise in the natural model equation. The noise component scatters the particle values with every resampling. Without this scattering component, the particle values at the maximum increase after every interaction. All particle values have the same values at some point due to the finite number of particles. The result is that the PDF function cannot be constructed by the particle. No resampling is implemented in M-LU. The issue with degeneracy is solved using new weights update strategy described in 8.4.4.

8.4.4 Particle update

The main goal of the update procedure is to evaluate new values for the weights using the measurement. The weights represent the PDF function. The updated coefficients are a multiplication of the previous weights with the conditional probability in the classical Particle filter.

$$w_k^{*i} = w_{k-1}^{*i} p(y_k | x_k^i)$$

Using multiplication has the advantage that it can be easily implemented and is not influenced by normalization (factorization). A disadvantage of the multiplication with a Gaussian importance function (conditional probability) is the extreme decrease of the coefficients at the low Gaussian values. The variance of multiplication increases with every interaction. This is also one of the reasons for the degeneracy of the coefficients.

The PDF of EI must be proportional to the PDF of the RUIs as already mentioned (notice not equal). Making use of this fact at the update, the multiplication can be exchanged by addition:

$$w_k^{*i} = w_{k-1}^{*i} + p(y_k | x_k^i)$$

The big advantage is that there is no degeneracy of the coefficients. The variance does not increase rapidly. The resulting function is proportional to the PDF and there is no degeneracy.

A major disadvantage of the method is the permanent commutation of the coefficient values. The values can only increase. Modern computers work with finite numbers, so the hardware will be overloaded at some point. Normalisation is a way of solving the increasing in the weights.

$$w^i = \frac{w^{*i}}{\sum w^{*i}}$$

Unfortunately, the normalisation changes proportionally between the updated coefficients. The prior updates become less important to all posterior following updates after normalisation. This follows directly from the mathematical equation for normalisation. For example: one weight after two normalisations at step $k-1$ and $k-2$ is:

$$w_k^i = \frac{w_{k-1}^i + p(y_{k-1} | x_{k-1}^i)}{\sum_i w_{k-1}^i} = \frac{\frac{w_{k-2}^i + p(y_{k-2} | x_{k-2}^i)}{\sum_i w_{k-2}^i} + p(y_{k-1} | x_{k-1}^i)}{\sum_i w_{k-1}^i} = \frac{w_{k-2}^i + p(y_{k-2} | x_{k-2}^i)}{\sum_i w_{k-2}^i} + \frac{p(y_{k-1} | x_{k-1}^i)}{\sum_i w_{k-1}^i}$$

The conditional probability at $k-1$ and $k-2$ is divided to different divisors, thus the updates are not considered fair.

There must be as little normalisation as possible in M-LU to avoid unfair treatment. The normalisation is carried out only when the coefficients reach the threshold to overload.

Using addition instead of multiplication increases the performance of the algorithm. There is no degeneration of the weights or the particles. The solution is acceptable, since the LU distribution procedure only requires proportionality to the PDF of EI. The effect of normalisation does not lead to underperformance, as the simulation shows.

An interesting fact is that normalisation on regular basis can be considered as a forgetting factor. This could have a positive influence on an environment with a dynamic PDF.

8.4.5 Particle moving

The particle positions (values) are very important for building the PDF. The particles must not be distributed in the same way as the PDF function since the information for the function is carried by the weights. There are two requirements for the particle positions. First: the number of particles must be sufficient to represent the PDF. Second: the particles must be distributed at the higher points of the PDF. The distribution of the particles is very important. For example: if the PDF has almost zero values up to 1000, than there is no sense in keeping positions at less than 1000.

With zero knowledge and no use of resampling, the particle positions are constant throughout the whole simulation. To overcome this shortcoming an algorithm has been developed for moving of particle. Moving is a synonym for changing the value and stressing that the old value is dropped.

If there is no particle value closer than x_{bin} to the measured value than the particle with the lowest weight is moved to the measured value. The weight of the moved particle is set to zero. The procedure is executed before updating the weights.

The algorithm moves the particle with the lowest weight to cover the PDF high values. The high values of the PDF are represented by large weights. Only particles with lower weights are moved. The distance size of x_{bin} is considered sufficient, since the LUs are uniformly distributed (constant) in the bin of x_{bin} .

8.5 Simulation

The main target of the simulation is to achieve proof of concept, thus to show the qualities of the new method. The results are compared with the currently used monotone constant interval update in a fair way. The analysis includes many parameters for the performance of the algorithm. The simulation shows the advantages and the practical difficulties. A benefit of the simulation is also the possibility to test the algorithm with different threshold values, so that the limits of the method can be found.

8.5.1 Simulation structure

The MATLAB 7 software is used for the simulation. The software is very suitable for this type of simulation. On the one hand, it provides many integrated mathematical functions: array sorting, histogram, approximation etc. On the other hand, it is a flexible programming language for creating non-standard functions. The simulations are done on PC hardware: AMD Athlon, 512 MB RAM and 1.15 MHz CPU.

The simulation is offline using the update/Event Intervals rather than absolute time, thus the clock is not used, see 8.5.2. The code efficiency is out of scope for this simulation. The stress is the accuracy of the predicted values, thus the minimum number of updates with smallest possible error (disconnection).

The simulation focuses on gathering statistical information for predefined Event Time Points (ETPs). Therefore, the structure of the code is inverse to the real implementation. The simulation code calculates for given Event Time Points the needed Update Time Points, Measured Intervals etc. The structure of the code is presented in Figure 8.8. It consist of six main blocks, their purpose and implementation are as follows:

Initialisation The semi-random intervals are generated in this step. Five representative semi-random cases are used in this thesis, see 7.5.1. The intervals correspond to the distance between the Event Time Points (ETPs). The Event Intervals (EI) are relative to the start of the filter cycle and the intervals are relative to each other. Intervals between the ETPs can be generated in advance. In contrarily, the start of the filter cycle depends on the quality of the estimation and cannot be pre-calculated. The Absolute EIs (AEIs) must be calculated systematically, thus cycle after cycle. The intervals between the ETPs are converted to EI using subtraction of remainder, explained later in this chapter.

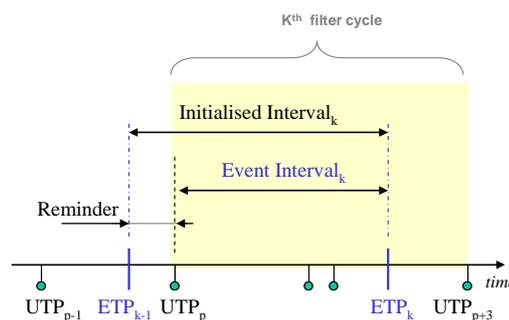


Figure 8.7: Reminder

Location Update measurement The procedure finds the Update Interval where the event occurs. Then it calculates the error, i.e. the disconnection time. The Maximum Disconnection Interval (MDI) is the size of the Update Interval (UI) with ETP. The estimated value is in the middle of the UI. The number of updates needed to detach the event are calculated in order to get the method qualities. All values are stored in an array for later statistical analysis.

A remainder is used to align the Event Intervals and Relative Update Intervals in the filter cycle. It converts the initialised intervals to EI. Basically, it is the difference between the initialised intervals, the ETP, and the following UTP, see Figure 8.7. It is executed before every filter cycle of prediction. Otherwise, the Event Interval is unknown in the filter cycle.

Update Procedure The particle weights are updated considering the new measured value in this block. The procedure is a straightforward implementation of the algorithm: addition of the weight with the conditional distribution expressed by Gaussian distribution.

Normalisation The weights are normalised if the maximum weight has reached a certain threshold value. The weights are divided to their sum, see 8.4.4.

Prediction If there is knowledge of the natural model (Hidden Markov Model), a prediction of the new particle position is carried out. In our simulation there is zero knowledge about the model, thus the predicted particle value is the same as the current one.

Analysis This part presents variables important for comparing the quality of the results. The four diagrams explained in 8.6 are used to present the results.

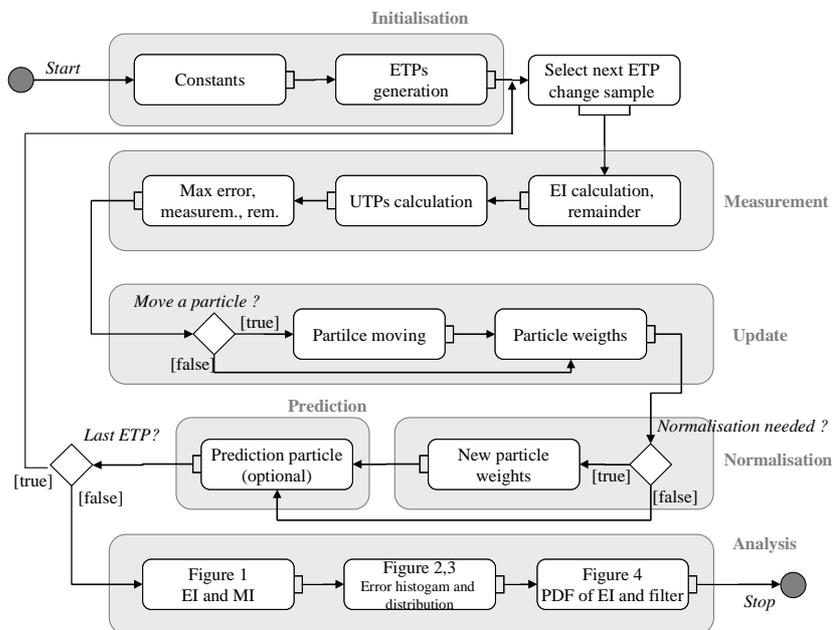


Figure 8.8: Simulation structure

8.5.2 Periodic behaviour dependent of the absolute time

The filter prediction mechanism uses only intervals between the events not considering the absolute time. The simulation code does not correspond to real implementation, where the absolute time can be considered.

The performance of the algorithm can become poor when the Event Time Points depend on the absolute time. The PDF cannot be built in the correct way, since it considers the intervals between the ETPs. For example: a mobile device owned by a daily worker. An active time is between 8:00h and 19:00h, where there are more events. There will be less Event Time Points after 22:00 to 6:00h. An inadequate PDF will be built if only the intervals are considered. The small size interval will dominate the PDF and small Update Intervals will be executed even in late hours. The performance of the algorithm will be even poorer than a constant interval since significant input is not considered in the simulation.

A simple transformation helps to handle events dependent of the absolute time. The absolute time dependencies are circular since the absolute time is circular. For example, there is a day cycle. There is no need to change the code of the simulation to implement a time clock. The Event Intervals are made relative to one anchor time point within the cycle, if there is a periodic dependency. The Event Intervals are no longer relative to the start of the filter cycle. If there is a day cycle, then the Event Intervals within one day are set relative to midnight (00:00h). In other words, the Event Interval is equal to the intervals between the start of the day and the Event Time Point. If there are PoA changes at 7:00h and at 7:10h, the Event Intervals have the length 7:00h and 7:10h. The events in following days are relative to the 00:00h of the day, in which they happen. The length of the intervals is from 0 to 23:59h. The relative time point must be in the cycle, i.e. day, but it must not be 00:00h. This principle works well with all time points in the day. Other points than midnight will only shift the PDF.

The Event Intervals can be handled properly using this simple pre-processing. The simulation works properly with cycle dependencies. The principle is the same for day, week, month etc. This requires knowledge of the cycle period in advance. This mechanism for a day is used in the simulation for simulating the real data.

8.6 Simulation results

The simulation is done with the input values generated in the same way, see 7.5.1, for all M-LU algorithms, thus Fuzzy controller und extended Kalman Filter. The representation and analysis is done as described in 7.5.2. In this way the M-LU algorithm can directly compared.

8.6.1 White noise

The intervals between the ETP are generated with the following equations:

$$aEI_r = N(0, \sigma) + 200$$

$$\sigma = 10$$

where aEI_r is the r^{th} Event Interval. $N()$ denotes normal distributed random values with parameters mean value zero (0) and standard deviation of 10. The aEI_r is shifted with 200.

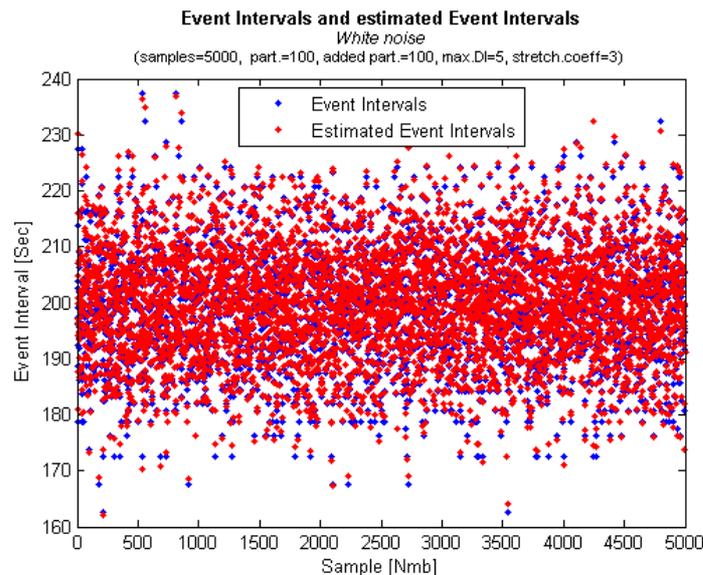


Figure 8.9: White noise EIs and estimated EIs

At the first Figure 8.9, the generated Event Intervals (EI) and estimated EIs are shown. This is the typical white noise distribution.

The summary of the result is presented in Figure 8.10. The blue bars show the histogram of the Update Intervals (UI) with Event Time Point (ETP). In the best case, the histogram must have high values close to zero. This means the Update Interval was small, where the event happens. Consequently, the disconnection intervals (DI) are small. The red dashed line shows the maximum defined DI (5 sec). The blue bar near the maximum DI shows that the prediction algorithm has failed and the Update Interval is the maximum possible. The bars at maximum DI must be as small as possible.

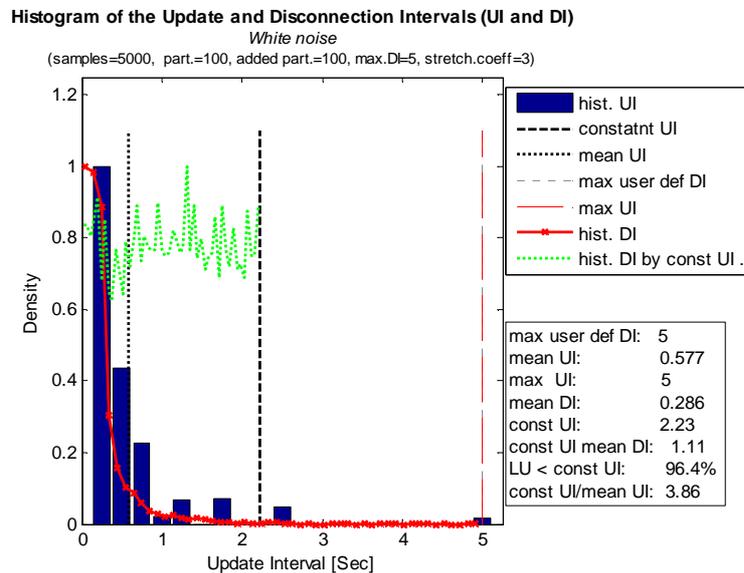


Figure 8.10: White noise results histogram

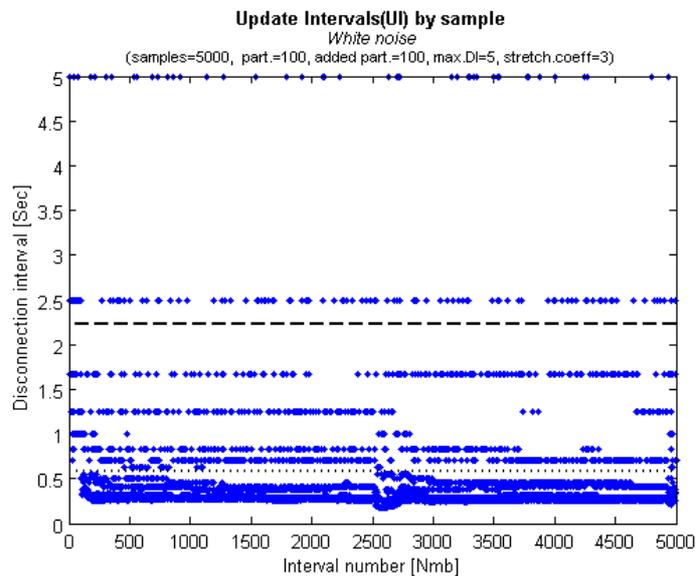


Figure 8.11: White noise, UI with ETP by sample

The green dotted curve shows the histogram of Disconnection Intervals when using the constant Update Intervals (const UI) with the same resources. The red line shows the histogram of the real DI by the new method.

As described in 7.5.2 the results are compared in a fair way with the constant interval update. This means the constant intervals are calculated with the same number of updates in the simulation time. The vertical dashed red line shows the constant UI.

The numerical results are shown at the bottom right corner of the Figure 8.10, for example the mean UI by the new method and by the const UI. The ratio between the two

values gives a simple qualifier for the performance. The ratio is 3.86. This means the new method significantly outperforms the constant UI by almost 4 times. Furthermore, 96.4% of the UI with ETP by the new method were smaller than the constant UI, thus almost all updates are better than with constant updates (Figure 8.10, $LU < \text{const UI}$). There are less failures of prediction, which is also acknowledged by the small size of the blue bar (histogram of UI) by the maximal DI by the user.

One big advantage of the new method must be pointed out: it keeps the same performance by increasing the Event Intervals with constant shift. Currently, the EI lies between 150 sec and 250 sec. If we shift the values between 1150 and 1250 the performance of the algorithm stays the same. In contrast, the performance of the constant Update Interval decreases since more updates are executed in vain. The outperformance of the new method increases by shifting.

Figure 8.11 shows the UI by sample. It can be clearly seen that UI in the first samples is not very good because the PDF is not yet estimated. The PDF generated by the Particle and the one estimated at the end of the simulation are shown in the following Figure 8.12. The PDF fits very well, which is also verified by the results.

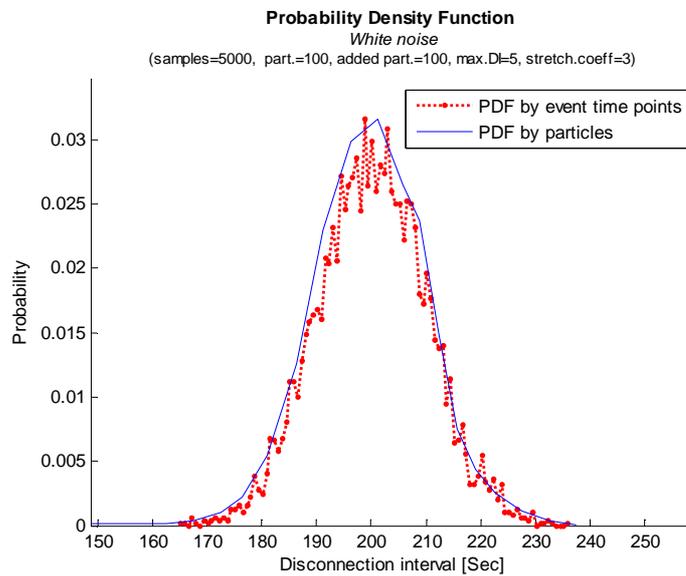


Figure 8.12: White noise, PDF by particle and estimated

8.6.2 Two rotating white noise sources

There are two white noise sources, which are rotating. They can be described by:

$$aEI_r = N(100, \sigma_1) \text{ or } N(100, \sigma_2)$$

$$\sigma_1 = 10, \sigma_2 = 10$$

The operator *or* denotes the rotation with 50% probability. Half of the values are generated by the first distribution and half by the second. The results are shown in Figure 8.13.

The histogram of the results in Figure 8.14 acknowledges the good performance of the new algorithm. The mean UI is 2.37 smaller than the constant UI. 94.7% of the updates are smaller than the UI by constant updated. It can be concluded that the new method performs independent from the PDF form, as designed (Figure 8.15).

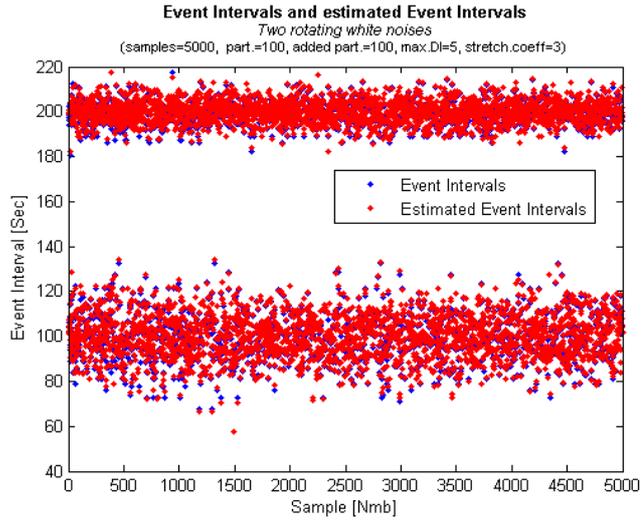


Figure 8.13: Two rotating white noise sources, EI and estimated EI

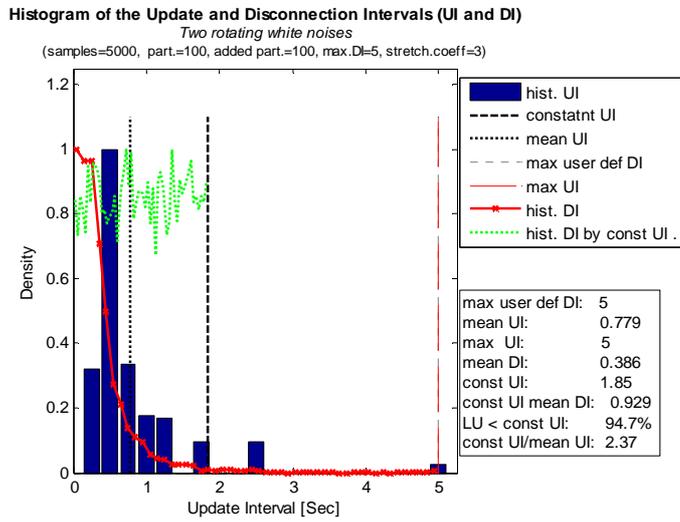


Figure 8.14: Two rotating white noise sources, histogram results

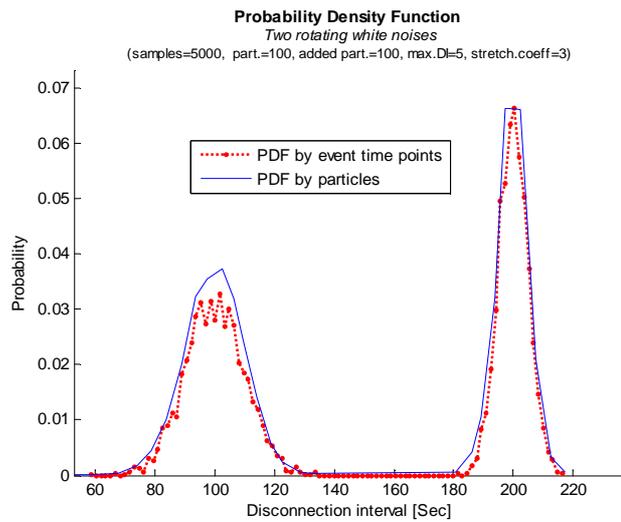


Figure 8.15: Two rotating white noise sources, PDF by particle and estimated

8.6.3 Sinus based EI with white noise

For this simulation, a sinus-based signal with white noise is used:

$$aEI_k = \sigma \sin\left(4\pi \cdot \frac{k}{S}\right) + \frac{7}{10} \sigma \sin\left(\frac{3}{2}\pi \frac{k}{S}\right) + \sin c\left(\frac{k}{S}\right) + N\left(500, \frac{\sigma}{2}\right)$$

The variable S indicates the total number of samples, S=5000. Here, k is the index of the calculated sample and σ is equal to 10. In Figure 8.16 the EI are presented. The Figure 8.17 presents the result histogram. The results show that this is an even better case for the new algorithm. The 97.8% of the updates are better then the constant Update Interval. The constant UI is 3.84 times bigger then the mean UI. This is generally due to the bigger values of Event Interval, compared to the previous simulations, the values lie around 500 sec.

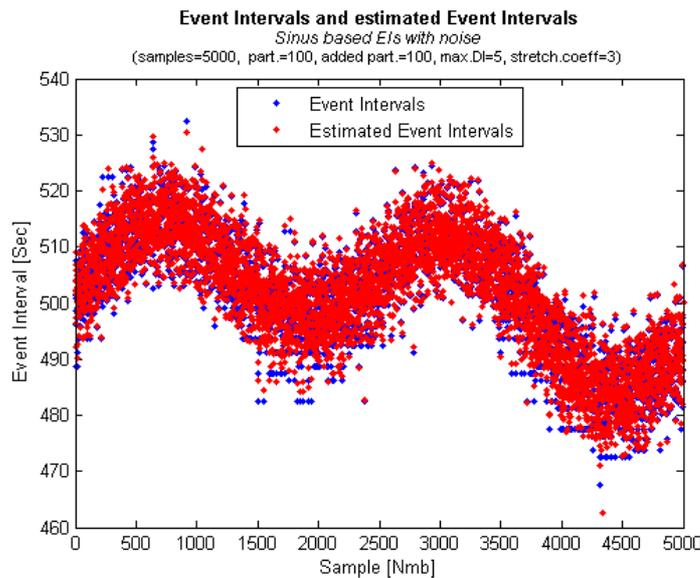


Figure 8.16: Sinus based, EIs and estimated EIs

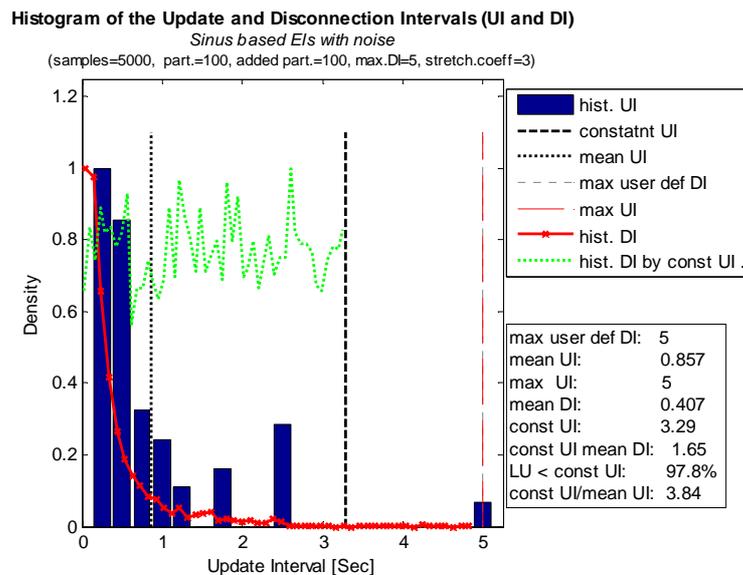


Figure 8.17: Sinus based, histogram results

8.6.4 Non linear EI with white noise

The Event Intervals are generated with recursion, where the next value depends on the previous one. The equation used in [11] and defined as:

$$aEI_k = \frac{aEI_{k-1}}{2} + \frac{25r_{k-1}}{1+r_{k-1}^2} + 8\cos(1.2k) + N(200,20)$$

The EI and the estimated EI are shown in Figure 8.18. The results in Figure 8.19 again show a very good performance of the algorithm. The mean UI is 2.24 times better than the UI with constant UI. Over 93.1% of the updates are better than the constant UI.

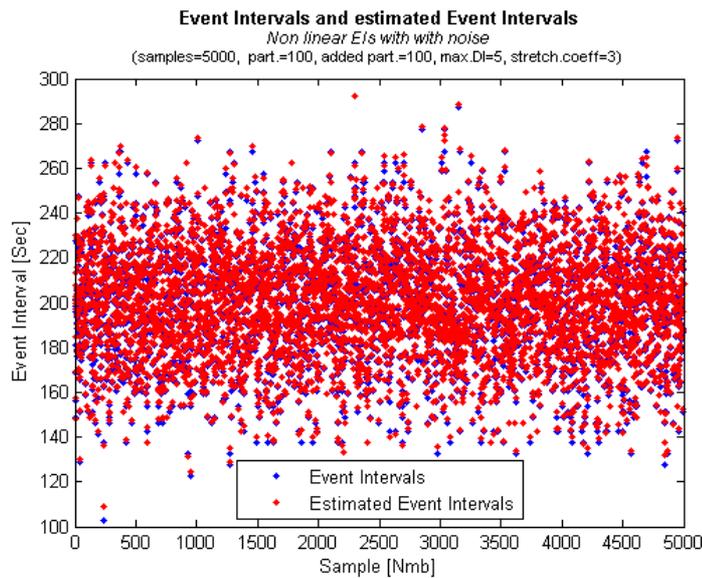


Figure 8.18: Non linear signal, EI and estimated EI

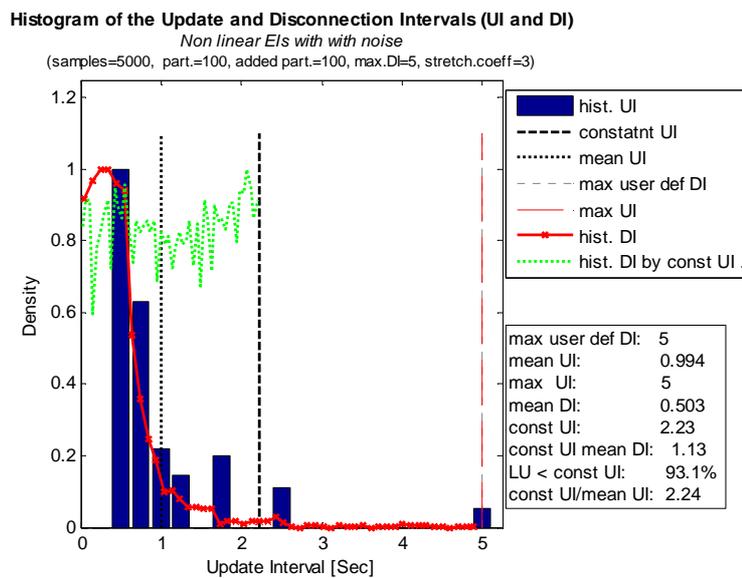


Figure 8.19: Non linear, histogram results

8.6.5 Real data

The algorithm was tested with real data. The data contains the login activities in 5 days of dial approx.1000 users. The EI is the time points when the network is accessed. More than 2500 EI were captured.

The users do not generate events fast so the input parameters of the method are increased, i.e. the maximum Disconnection Interval is to 300 seconds (5 min), EIs are 2000, there are 2000 particles, 500 added updates, and stretch coefficient of 10. The parameters are fit manually.

Further important optimisation is the assumption that the data is based on a day cycle. As described in chapter 8.5.2. The EIs are set relative to midnight (00:00h) of the same date. Without this pre-processing, very poor results are achieved. The PDF within one day is shown in Figure 8.22. The typical working cycle with low activity in the night and high activity during the day is clearly shown.

The results in Figure 8.20 show that the mean UI by the new method and the constant UI are almost the same. The 77.3% of the updates are better then the constant UI. The Figure 8.21 presents the UIs with ETP by sample.

The input data represent a difficult test case. The outperformance is not as good as with the generated EIs. The main difficulty is that the assumption of the day cycle is partially right. Additionally, there are also week, month etc. cycles. All these are overlapped, where the day cycle has ca learly dominating role.

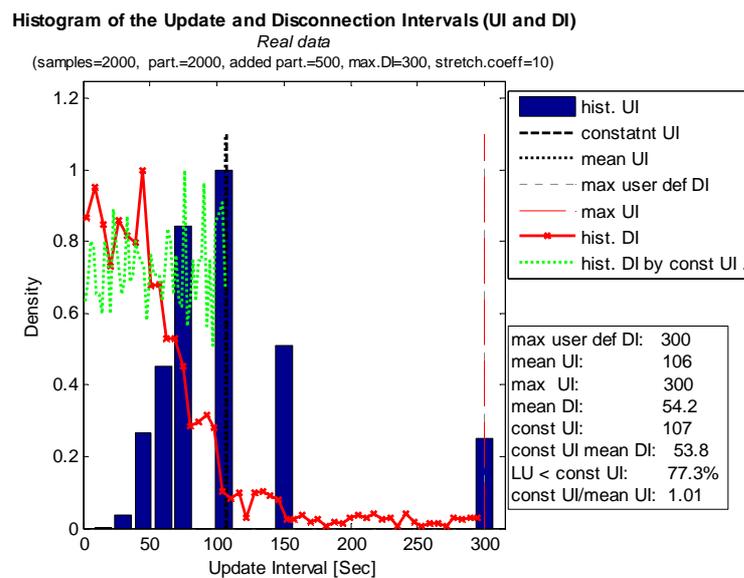


Figure 8.20: Real data, histogram results

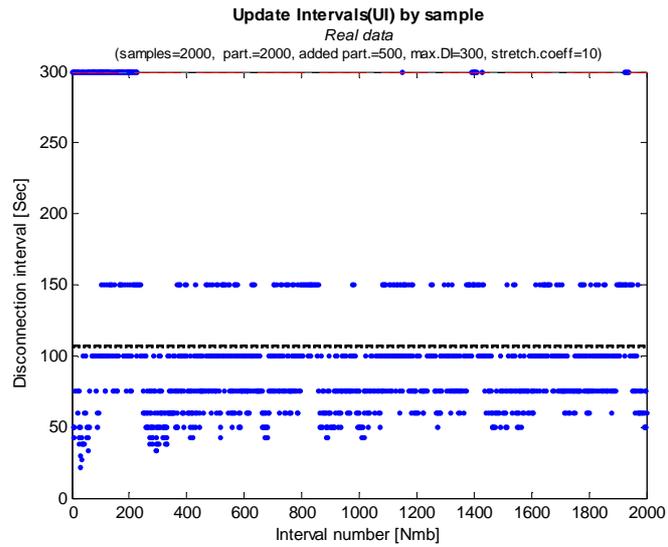


Figure 8.21: Real data, UIs with ETP

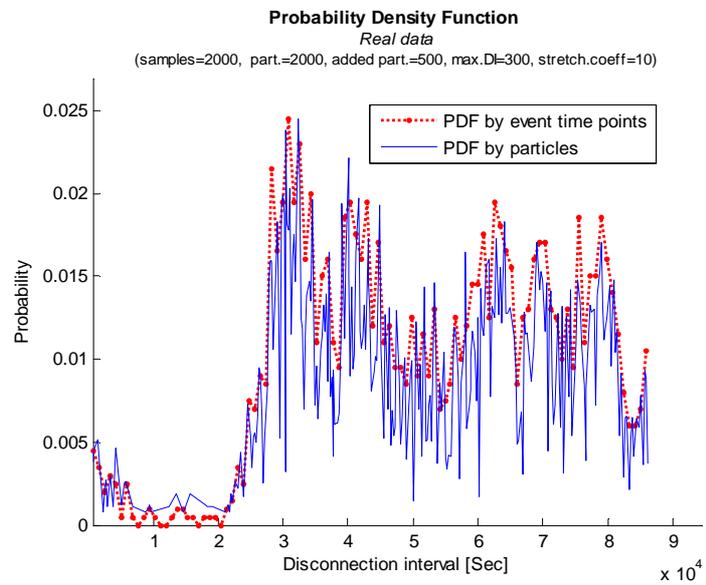


Figure 8.22: Real data, PDF

8.7 Conclusion and future work

The Particle filter has advantages over the constant interval trough using the history of the PoA changes. The performance of the Particle filter stays constant, independent of the PDF curve, noise, offsets etc. The algorithm requires sufficient samples to estimate the PDF. Although Monte Carlo is a suboptimal method, it delivers very good results for non-linear systems.

The simulation clearly shows the advantage of the new algorithm. In all cases, there was better performance than the constant UI. There are still areas in which the algorithm can be improved. Future work could contain the following optimizations, which are out the scope of this thesis:

- The parameters, such as the number of particles, added updates etc are chosen manually. The values for the simulation are selected through tests. Optimisation of the parameter must be implemented in real deployment. This is automatic control engineering task and out of scope in this document.
- The cycle events, such as day, week etc, are not considered in overlay. Currently the algorithm can apply only one of them. For example: day, week or month. It must be enhanced to consider more of them in an overlaying manner.
- The PDF can change over the time. The algorithm will slowly adapt the PDF by every new EI. A forgetting factor can be added to the PDF to support fast changing PDFs.

8.8 References in chapter 8

- [1] Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems", ASME, 1960
- [2] Gordon, N., Salmond, D., and Smith, A. F. M., "Novel approach to nonlinear and non-Gaussian Bayesian state estimation", Proc. Inst. Elect. Eng., F, vol. 140, pp. 107–113, 1993.
- [3] Arulampalam, M. S, Maskell, S., Gordon, N., and Clapp, T., "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking", IEEE, VOL. 50, NO. 2, 2002
- [4] Doucet, A., Godsill, SJ and Andrieu, C., "On Sequential Monte Carlo sampling methods for Bayesian filtering", Stat. Comp., 2000
- [5] Copsy, Keith, "Tutorial on Particle filters", Pattern and Information Processing Group,DERA Malvern
- [6] Gordon, N.J.: Lake Louise, "Presentation General PF discussion" , October 2003
- [7] Fearnhead, Paul, Merton College, "Sequential Monte Carlo methods in filter theory",University of Oxford, PhD Thesis, 1998
- [8] Bergman, Niclas, Linköping, "Recursive Bayesian Estimation Navigation and Tracking Applications", Studies in Science and Technology. Dissertations, No. 579
- [9] Smith, A. (Foreword), A.Doucet (Editor), N.Freitas (Editor), N.Gordon (Editor), "Sequential Monte Carlo Methods in Practice", Springer; 1 edition (June 21, 2001)
- [10] Forsyth, D.A. and Ponce, J., "An introduction to probability", draft version of "Computer Vision - A Modern Approach", Prentice-Hall, 2002.
- [11] Forsyth, D.A. and Ponce, J., "Tracking with non-linear dynamic models", draft version of "Computer Vision - A Modern Approach", Prentice-Hall, 2002.
- [12] Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks", K.Murohy, S.Russel, Springer, 2001
- [13] Casella, George, Berger, Roger L., , "Statistical Inference", Duxbury Press , 2 edition , June 18, 2001
- [14] Liu J. S. and Chen R., "Sequential Monte Carlo methods for dynamical systems", Statist. Assoc., vol. 93, pp. 1032–1044, 1998.
- [15] Tzvetkov, Vesselin,"Optimization of mobile updates using Particle filter", IEEE ChinaCom, August 2008

9 Mobile Location Update protocol with Adaptive Fuzzy controller

This chapter describes an adaptive Fuzzy controller for realisation of the Mobile Location Update protocol. This is an alternative solution with Fuzzy controller to the ones described in chapters 8 and 10.

The section 9.2 gives a brief overview of Fuzzy system in order to align the readers and clarify the notations. A basic understanding of Fuzzy logic is required by reading this. The deployment of Fuzzy controller in M-LU is presented in 9.3. The controller predicts the next Event Interval. The Update Time Points are calculated using a transformation function described in 9.3.4. This function converts constant intervals into non-linear Update Intervals proportional to PDF of Event Time Point. The controller rules are generated with One Pass (OP) training and optimised with Recursive Least Squares (RLS) algorithm.

The simulation program structure and implementation are described in 9.4. The simulation provides proof of concept, where the properties of the method can be verified. The simulation results are presented in 9.5. In order to compare the performance of the methods, the input's equations are the same as in the Sequential Monte Carlo method (chapter 8) and extended Kalman Filter (chapter 10).

9.1 Contributions

The main theoretical contributions of the author in this method are: (1) First, the creation of a model for the Mobile Node behaviour on which an adaptive Fuzzy controller can be deployed, see 9.3. (2) Second, the position of Update Time Points is calculated with a novel transformation function, see 9.3.4. (3) Third, handling periodic ETP dependent on absolute time, see 8.5.2. The practical contribution is the simulation of the new method and comparing the results to the classical constant update method in 9.4.

The novel method with the simulation results has been published and presented by the author at IEEE conferences [9, 10]. The method can be applied to other protocols as shown in chapter 11.

9.2 Fuzzy Logic Systems

This chapter gives an overview of Fuzzy controller. There are two types of descriptions when exploring a phenomenon: objective and subjective. The objective knowledge creates a system model with univalent mathematical equations based on known physical properties. A typical example are engineering solutions using the dynamics of a falling body. The majority of engineering implementations are based on objective knowledge since the mechanisms and the relations have been very well studied in classical disciplines, thus mathematics and physics. The second possibility for system description uses subjective knowledge representing expert experience. The subjective description cannot be expressed with exact values but only by human understandable quantities. For example: "it is quiet early in the morning". The subjective description of the problem is usually gathered over a long time and is more intuitive than exact.

The objective description has the advantage of delivering precise values. It can be processed by computers and furthermore it is universal i.e. no matter in which country and by which person the result of the procedure will be the same. All calculations are obtained from strict mathematical definitions. For this reason, it is popular for solving engineering problems.

The subjective knowledge, in contrary to objective knowledge, is based on classification and expression. There is a large scale of possible interpretations in the implementation. For example, the sentence “it is quiet early in the morning”. There are many possibilities for understanding this expression and it is difficult to conclude something for practical implementations. Still such expressions contain valuable information for solving the problem. The description is gathered with human experience and is known as expert knowledge. When dealing with complex problems where solutions cannot be found with objective methods, the subjective solution is a reasonable alternative. The subjective knowledge is often the only alternative, like for example in historical descriptions.

The discipline for defining, creating operations and processing a subjective knowledge is called Fuzzy logic. It defines rules for working with multivalent values, which is not directly possible in the classical mathematics.

9.2.1.1 Description of Fuzzy logic systems

This chapter gives a very brief overview of Fuzzy system. More details can be found in [1, 2].

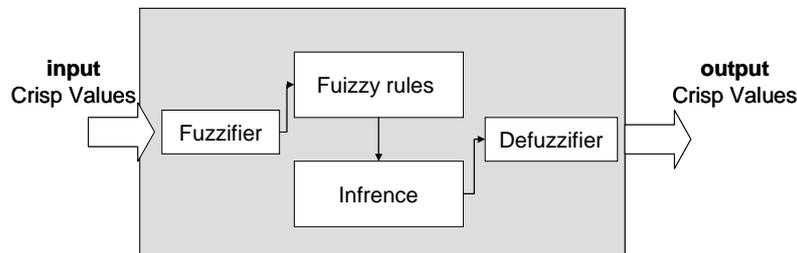


Figure 9.1: Fuzzy system

Fuzzy systems consist of the following blocks: fuzzifier, Fuzzy rules, inference and defuzzifier, see Figure 9.1. The crisp input values are transformed by the fuzzifier in a Fuzzy set. The Fuzzy set extends the classical set theory by adding assessment belonging. The Fuzzy set consists of elements with a membership function. The function value acts as an indicator of belonging to some element. The membership function is denoted as μ and has values in the interval 0 to 1, thus $\mu_x \rightarrow [0,1]$. Then the Fuzzy set \tilde{A} of value x in space of X is defined by:

$$\tilde{A} = \{(x, \mu_A(x)), x \in X\}$$

A typical set can be the water temperature for example: hot with membership of 0.3 degree, cold with 0.2 degree, normal with 0.5 degree. The water temperature belongs to different degree to different elements. Multivalent values are not possible in the classical mathematic, where the water temperature can be 27 degrees for example and this is an exact crisp value.

The membership function can have a different form. It should represent in the best way the belonging to an element. Some popular membership curves are shown in Figure 9.2.

The membership functions can be: triangular, Gaussian bell, trapeze, single point or any other. Depending on the design, the form can be chosen freely. The most popular is triangular membership function, since it is easy to calculate and it delivers smooth transaction between the values.

A *triangular* membership function is defined by:

$$\mu_x(x) = \max\left(0, 1 - \frac{|x_i - x|}{c}\right),$$

where x_i denotes the maximum of the triangular at the abscise i.e. mean point. The constant c defines the spread of the membership function. The higher the c value, the more uncertainty there is. Another important membership function is *singleton*. It is defined as: $\mu_x(x) = 1, x = x_i$ and $\mu_x(x) = 0, x \neq x_i$. The advantage of *singleton* membership function is its simplicity, when providing defuzzification [2]. The opposite *nonsingleton* membership function has no zero values at more than one point: $\mu_x(x) = 1, x = x_i$ and $\mu_x(x) \geq 0, x \neq x_i$. The triangle membership function is *nonsingleton* according the definition.

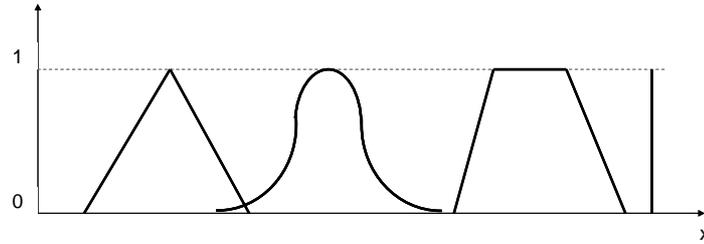


Figure 9.2: Membership functions

The crisp inputs are processed through the fuzzifier. Then the sets are evaluated using the Fuzzy rules with inference. In every Fuzzy system, there is an associated set of rules. The rules represent the linguistic interpretation of the system behaviour. The rules are the result of exploring the system over the time. Each rule expresses a sentence of the experience (expert knowledge). The rules have the form:

$$R^l : \text{IF } u_1 \text{ is } F_1^l \text{ AND } u_2 \text{ is } F_2^l \text{ AND } \dots \text{ AND } u_n \text{ is } F_n^l \text{ THEN } o^l \text{ is } G^l,$$

where R^l is l -th rule, u is the input variable, F is element of the Fuzzy set, o is the output and G is an element of the output Fuzzy set. There is an antecedent part and a consequent part. The antecedent part may consist of multiple inputs (u). The consequent part has one output variable (o). Multiple consequent variables can certainly be implemented, when the multiple variables are subsets of a single consequent set. A single consequent variable is considered further in the text.

The different antecedent membership functions are joined with logical AND or OR. The logical operations in Fuzzy logic are defined differently from the classical mathematical understanding. All *T-Norm* operations can be used for the definition of AND and OR. The *T-Norm* (t-norm) operations are simple relations between values [3]. In most of the Fuzzy implementations the AND of variables means the minimal of these. The maximum variables are implemented in OR operation.

The resulting Fuzzy elements are aggregated in one set using *T-Conorm* operation [3]. In practice, T-Conorm operation is the maximum of all input elements. Let o be the output of a single rule and O be the output the Fuzzy set. It can be denoted by:

$$O = \bigwedge_{i=1}^M o_i,$$

where \bigwedge is the T-Conorm operation and M the number of rules. Expressing the consequent set with the antecedent part gives:

$$O = \bigwedge_{l=1}^M \left[\mu_o^l(y) * \bigwedge_{k=1}^n \mu_{Q_k}^l(x_{k,\text{sup}}) \right],$$

where μ_o and μ_Q is the membership functions of consequent and antecedent. The Operation $*$ is a single T-norm operation. The Operator T denotes the sequence of T-norm operations in the antecedent part of the rules. The system has n input variables. The centre of the consequent membership function is y .

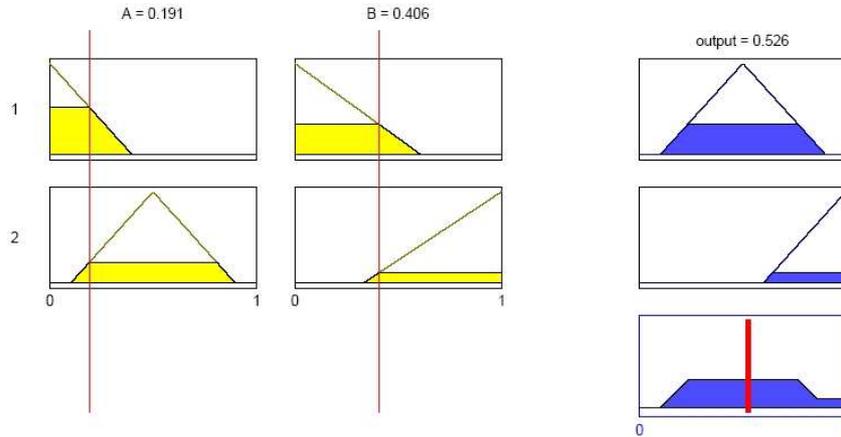


Figure 9.3: Fuzzy rules example

This Fuzzy set output must be transferred into a crisp value by the defuzzifier. Let the function $D()$ express the defuzzifier. The most popular methods are Centre of Gravity (centroid) and Height defuzzifier. The centre of the mass is the crisp output value in the centroid method. In the second approach, the crisp output is the highest values. The centroid method favours the output of greatest area. The height method favours the rule with the greatest output value.

The output of crisp value is written as:

$$f = D(O) = D\left(\bigwedge_{l=1}^M \left[\mu_o^l(y) * \bigwedge_{k=1}^n \mu_Q(x_{k,\text{sup}}^l) \right]\right)$$

Let us define that the centroid method is used for defuzzification. Furthermore, the singleton membership functions are used for the rule output. The singleton has little deflection by large number of elements when compared to nonsingleton. The output can be expressed using these two definitions, as:

$$f = \frac{\sum_{l=1}^M y^l \left[\bigwedge_{k=1}^n \mu_Q(x_k^l) \right]}{\sum_{l=1}^M \bigwedge_{k=1}^n \mu_Q(x_k^l)} = \sum_{l=1}^M y^l p_{fb}^l(x) \quad \text{Equation 9.1}$$

$$p_{fb}^l(x) = \frac{\bigwedge_{k=1}^n \mu_Q(x_k^l)}{\sum_{l=1}^M \bigwedge_{k=1}^n \mu_Q(x_k^l)}$$

The $p_{fb}^l(x)$ is called Fuzzy Basis Function [4]. It will play an important role in the creation of the Fuzzy controller. The last two equations are central to the evaluation of the Fuzzy controller with the method proposed here.

This classical Mamdani Fuzzy model is shown in Figure 9.3, where A and B are inputs and the membership functions are triangular. The crisp values for the input are A=0,191 and B=406. The crisp output is 0.526 calculated with centroid method, shown in the figure with a thick red line.

9.2.2 Creating Fuzzy rules with training methods

The Fuzzy rules express the expert knowledge on a certain phenomenon. In many cases, there is insufficient information to create exact membership functions and rules. Consequently, acceptable results cannot be achieved. If actions-reactions of the system have already been observed, then they can be used to create Fuzzy rules and adjust the membership functions. Applying known system input and output values to adjust the values of the controller is called *training*.

The training data should include all possible system states and changeovers in the best case. Poor performance is the result when some system states are not included in the training data.

9.2.2.1 One Pass method

One Pass [5] is a simple and fast method for creating Fuzzy rules using known input-output pairs. Let the observed value pairs be $(x_1^{(1)}, x_2^{(1)}, \dots, x_n^{(1)}, y^{(1)}), (x_1^{(2)}, x_2^{(2)}, \dots, x_n^{(2)}, y^{(2)}), \dots$, where x is the crisp input value and y is the output value. There are n input variables and single output variable.

- 1) The intervals of input and output variables are expected and set in $[x_1^-, x_1^+], [x_2^-, x_2^+], [x_3^-, x_3^+], \dots, [x_n^-, x_n^+], [y^-, y^+]$. The intervals can be adjusted dynamically, but are initially constant for clarity. Each interval is divided into M_i regions, where the number of regions can be different for every variable. The number is dependent on the required precision for the variable. Variables that are more important should have more regions. A membership function is assigned to every region. The membership function is greater than zero within the interval and zero outside the interval. Every membership function can be labelled, for example: very small, small, normal, big, very big etc. The label is to facilitate the verbal association. The exact form of the membership function is freely chosen. In this thesis, the triangular form is used because of the smooth transaction between the values.
- 2) Every pair of input-output values is evaluated against the membership functions. The input-output values are assigned to the region with the highest membership function. For example: a certain value has a maximum degree of membership when assigned to the "very small" element. One rule is created for every training pair in this way.
- 3) The degree of strength is calculated for every rule, in order to resolve conflicting rules. The degree D_l of l^{th} rule is calculated as follows:

$$D_l = \mu_y(y^l) \prod_{i=1}^n \mu_{x_i}(x_i^l),$$

where $\mu_y, \mu_{x_1}, \dots, \mu_{x_n}$ are the membership functions for the rule.

- 4) Conflicting rules have the same antecedent memberships, but different consequent membership. One rule must be selected for a certain combination of antecedents, thus only one output is possible for a certain combination of input values. If there are conflicting rules, only the rule with the highest degree is kept and the others are discarded.
- 5) After evaluating all values as described in 2) and 3) the rules base is ready for use. The new input values can be evaluated in the rule.

The One Pass (OP) method is widely used because of its simplicity and good results. Typical, a large number of training data must be processed before starting to use the rules. We recommend the use of the OP method together with rules created by the experts. The pre-defined expert rules can be used as a major part and the OP method can be involved for fine-tuning and supplementary rules. Simulation results for the OP method can be found in [2].

9.2.2.2 Recursive Least Squares method for optimisation

A major drawback of the OP method is the centres of the membership functions. These values are pre-defined without knowledge of the system. The optimal centres require prior knowledge of the variables distribution. The distribution is unknown at 1) of the OP method. The membership functions may misinterpret important data, since the centres must be distributed at the high values of the Probability Distribution Function.

The Recursive Least Squares method (RLS) is used to adjust the centre of the output membership values assuming, that the membership functions are symmetric over the mean value. Optimising the consequent centres leads to faster convergence to the real PDF of EI. The input rules are not changed.

The main idea of the RLS method is minimising the sum of square error. The error is the difference between the crisp Fuzzy output and the training data. It is denoted as j_{RLS} . Let d_i be the training data, $f(x_i)$ be the Fuzzy controller function and t be the number of training samples. The error is:

$$j_{RLS} = \sum_{i=1}^t \lambda^{t-i} [d_i - f(x_i)]^2$$

Here, λ is the forgetting factor with values $[0,1]$, where 1 means considering all past values and 0 means forgetting all past values. The x_i variable represents the input variables in our case. The main goal of the RLS method is to minimise j_{RLS} with respect to $f(x)$. The $f(x)$ is dependent only on the means (centres) of the output membership functions. The $f(x)$ is according to Equation 9.1:

$$f(x) = \sum_{l=1}^M y^l p_{fb}^l(x) = Y^T P^{fb}(x),$$

where Y and P are the matrixes. If there is a training sequence of t values, then the matrix P has t rows (training data) and M columns (number of rules). Y is a matrix with a column of t training values. Using matrixes simplifies the notation and the processing. The RLS algorithm can be applied to the equation since $f(x)$ is the linear function with respect to Y . The recursion for Y is obtained by minimising j_{RLS} . The input-output pairs following the recursion must be evaluated for every training data set k [6]:

$$P_{k+1} = P_{k+1}^{fb,*}$$

$$G_n = \frac{L_k P_{k+1}}{\lambda + P_{k+1}^T L_k P_{k+1}}$$

$$e_{k+1} = d_{k+1} - P_{k+1}^T Y_k$$

$$Y_{k+1} = Y_k + G_k e_k$$

$$L_{k+1} = \lambda^{-1} L_k - \lambda^{-1} G_{k+1} P_{k+1}^T L_k$$

The $P_{k+1}^{fb,*}$ denotes the Fuzzy basis function at the k^{th} training pair. It contains a column with M values, one for each input. The G_k is the gain vector at step k . L_k denotes the inverse correlation matrix at step k . L_k is $M \times M$ and is initialised with the identity matrix. The value e_k is the priory error. Y is the matrix with the centres of the consequent membership

function. It is initialised with M zero values. More background information can be found in [6].

After evaluating all training sets sequentially, Y contains the centres of the consequent membership functions. The centres are chosen to minimise the j_{RLS} values. A great advantage of the RLS method is not finding some local minimum, but the function minimum [6].

The same principle can be applied by adjusting centres of the input membership functions. The RLS method optimises the membership centres of already developed Fuzzy rules and it is used in the simulation.

9.2.2.3 Other training methods

There are different methods for optimising the rules, such as Back-Propagation, Singular Values and Least Mean Square. They minimise different errors and therefore have different qualities. The performance and comparisons can be found in [4, 1, 2]. The minor differences in the algorithms are not decisive for the M-LU procedure. The author believes, without insisting on this, that RSL method performs very good in this framework. Certainly, other methods can be used without changing the concept. In this thesis OP is used with the RSL method.

9.3 Fuzzy controller for the Location Update procedure

In this section 9.3, the implementation of Fuzzy controller in the M-LU is described. The general idea is to create a Fuzzy controller for predicting the Event Interval (PoA change) using the past Measured Intervals (MIs). The controller works as black box. The input is several past MIs and the output is the prior estimated EI. This is the common method for predicting chaotic time series [2].

The Update Time Points are calculated from the prior estimated EI and the Maximum Disconnection Interval (MDI) according to 9.3.4.

The Fuzzy controller uses the intervals rather than absolute time values as already discussed in 7.3. The Event Intervals can be set relative to some point in the period if there are periodic events, as described in 8.5.2.

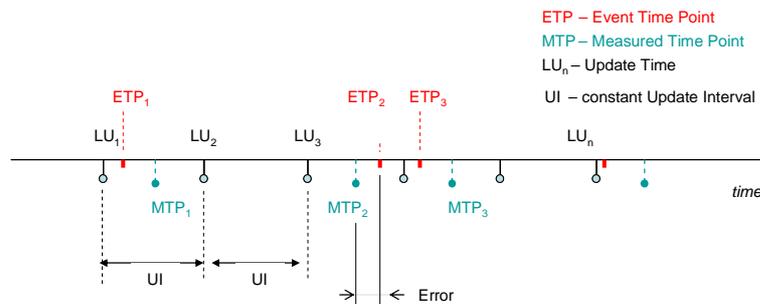


Figure 9.4: Training intervals

The membership function must be defined first, as described in 9.3.1. Then Fuzzy rules must be using expert rules in combination with training data see 9.3.2. Training data is collected during the training period using constant Update Intervals equal to Maximum Disconnection Interval (MDI). These intervals are sufficiently small to satisfy the application requirement and give good precision. The posterior estimated Event Time Point (MI) is assumed to be in the middle of the Update Interval to minimize the absolute estimation error. The MIs build the training pairs described in 9.3.2. The following Figure 9.4 shows all elements again.

After the rules are created, the centres of the consequent membership function are updated according to the OP method, see 9.2.2.1 and 9.2.2.2.

9.3.1 Defining the interval ranges and membership functions

The range of possible EI values must be defined before starting to train the controller, thus minimum and maximum values. When an EI is out of min-max value, it cannot be predicted. The predicted value will be at the extreme of the domain, but still within it. Increasing the difference between the defined domain and real values increases the prediction error. On the other hand, unnecessarily large intervals require more membership sets to achieve sufficient precision of the controller. More membership sets require more Fuzzy rules to cover all possible value combinations. More rules means more CPU and memory resources at the device. To recapitulate, the value range cannot be increased to infinite. An interval between 0 and 24 hours is defined in M-VPN, which seems to be a good value. The domain (range) should not cover all Event Intervals, but it should cover the majority of them. The output and input domain are the same in M-LU because the input and the output are the same variable types.

Each domain must be divided into regions of the membership functions. The number of regions defines the precision of the prediction. The higher the precision, the more regions are needed. The Maximum Disconnection Interval (MDI) defines the precision necessary to some degree. The membership region is proportional of the maximum disconnection. The membership functions overlap by 50% on the left and right side. An overlapping is very important since it covers directly not considered values combinations. The total number of regions in the input-output domain is:

$$N_{\text{number of regions}} = \frac{2 \cdot D_{\text{domain size}}}{R_{\text{region size}}},$$

where $D_{\text{domain size}}$ is the domains size, $R_{\text{region size}}$ is the region size proportional to the maximum Disconnection Interval. The region size is set 3 times the MDI in this simulation. This value is chosen through multiple simulation for delivering good results. It is an optimization task to find optimal values and it is out of scope in this thesis.

The antecedent membership functions are triangular. The advantages of the symmetric triangular form are the ease of calculation and good performance.

9.3.2 Creation of rule base

The expert rules provide raw information on how the controller should act. For example: they can be created using the assumption that the M-LU is implemented in the device used by an employee. As generally known, there are two maximums in business days: in the morning and after lunch. The maximums are higher on Monday than on Friday. There is a reduced level of activity at the weekend. Typical diagrams for day and week activity can be seen in Figure 9.5. Since there is no real data real data of mobile usage available, the data has been gathered from the Internet exchange point - PARIX. The Mobile Client movements depend on the usage, like smart phone, train etc.

Abstracting rules can generally describe the day and week activity and can be involved in covering the main extremes. They could consist of two rules, for example:

IF 7:00 h < time AND time < 20:00 h THEN Location Update Interval IS small
 IF Monday < day AND day < Friday THEN Location Update Interval IS small

In the next step, the rules are created through training as described in chapter 9.2.2.1.

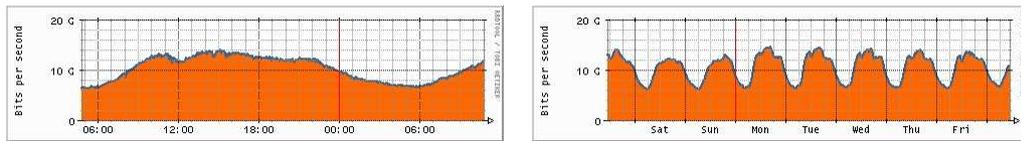


Figure 9.5: Day and week activity

9.3.3 Number of inputs of the Fuzzy controller

Before starting the OP (One Pass) method, the number of inputs of the Fuzzy controller must be set, i.e. how many past values must be considered in the prediction. Increasing the number of inputs improves the prediction precision. This can be compared with the interpolation of curves using more given points. The interpolation becomes more realistic when involving more points.

Unfortunately, increasing the number of inputs requires more rules to cover transition cases. The exact number of rules cannot be determined, since the form of the interval transition plays a major role. It is factorial incensement of the number of inputs according to the authors subjective experience. In the simulation, about 100 rules were required for domain 0 to 700 sec with 3 input variables. Using more than 10 past input values requires many thousands of rules.

The input values (Measured Intervals) are typical sequentially following values, thus values with index $k, k-1, k-2, \dots, k-n$ where k is the current index and n is the number of inputs. Reducing the number of inputs reduces the past time period considered in the controller. Reducing the considered past values decreases the prediction ability. To overcome this problem, not sequential following sequential values can be involved. For example series such as $k, k-1, k-3, k-5, k-10, \dots$. The considered period increases and the precision decreases. It is a trade off between the precision and the considered past values. The use of not sequential following past values is only mentioned and it is part of future work. M-LU uses monotone following values.

After the rule base is created, the centres of the consequent membership function must be updated according the RLS optimisation method. The implementation is straightforward from the theoretical part, see 9.2.2.2.

9.3.4 Distribution of the Update Time Points

The Fuzzy controller predicts the coming Event Interval using past Measured Intervals (MIs). The predicted Event Interval (EI) shows the highest probability where the event (PoA change) could happen. The Fuzzy controller does not deliver the PDF of the EI but only a single value in contrast to the Particle filter described in chapter 8. The PDF cannot be constructed from this single value. This is a general issue of the Fuzzy method. The requirement of the chapter 7 cannot be met directly, i.e. the Update Intervals (UIs) to be distributes as the real PDF. This shortcoming is solved with the assumption that the probability of EI is in approximately Normal distributed with the standard deviation proportional to the Update Interval (UI). This principle is also applied in the extended Kalman Filter for M-LU presented and derived in 10.3.

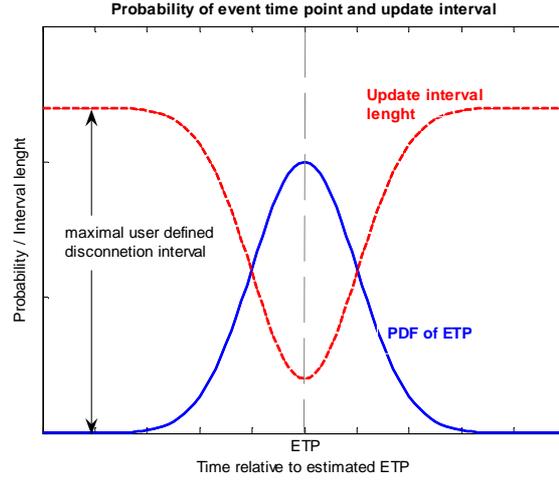


Figure 9.6: Location update interval and the probability of PoA change

Normal distribution of EI is continuous and not zero for all values. Implying this would mean that there is a nonzero probability at all time points, even in the past. This is certainly not the case. To overcome this controversy, we define the first condition: the probability for ETP is Normal distributed except for the past points. Using subjective knowledge, it is defined that the distribution must have very low values at the start of the filter cycle (zero time). The standard deviation is three times of the predicted EI as described in chapter 10.3.

The second condition is that the size of the Update Interval(UI) must also not exceed the Maximum Disconnection Interval (MDI) defined by the application.

An abstract curve of UI satisfying both conditions is shown in Figure 9.6.

Setting the UI proportional to the Normal distribution (PDF of EI) and not exceeding MDI is derived in chapter 10.5. The main idea is to create a transformation function for converting constant Update Intervals in a non-linear interval proportional to the PDF of EI.

The transformation function has an input the time point of the constant updates relative to the filter cycle. It depends on the parameter: the prior estimated EI (predicted EI) and the Maximal Disconnection Interval (MDI). The output of the function is Update Time Point considering the PDF of EI.

The transformation function $T()$ derived in chapter 10.6 is:

$$k = \frac{P_k^+}{\text{erf}^{-1}(0.95)}, \quad u = \frac{k\sqrt{\pi}}{2} e^{2\text{erf}^{-1}(0.95)}, \quad C_2 = 0.95 \cdot \frac{\sqrt{\pi}}{2} e^{\text{erf}^{-1}(0.95)} - P_k^+$$

$$T'(x) = \begin{cases} u \cdot \text{erf}(x/k) & x \in [-l, +l] \\ ax + C_2 & x \in (-\infty, -l) \cup (l, \infty) \end{cases}$$

Then the inverse function is:

$$T'^{-1}(x) = \begin{cases} k \cdot \text{erf}^{-1}(x/u) & x \in [T'(-l), T'(l)] \\ x - C_2 & x \in (-\infty, T'(-l)) \cup (T'(l), \infty) \end{cases},$$

where

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

The P_k^+ is the prior estimated Event Interval (PoA change interval) and x time point of the constant updates. An example of the function is shown in Figure 9.7. The detailed derivation is presented in 10.3.

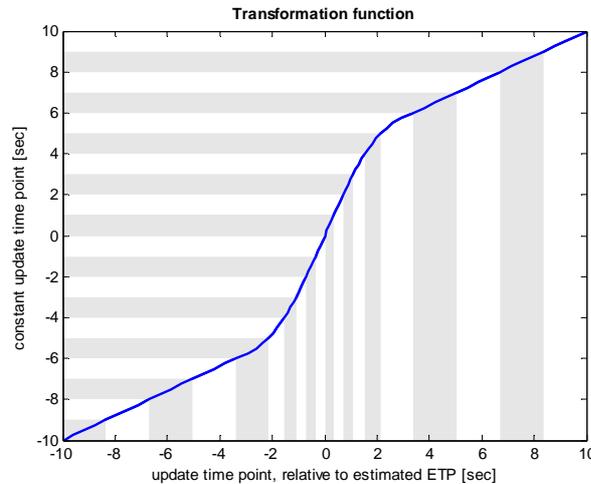


Figure 9.7: Transformation function

9.3.5 Implementation procedures and structures

An overview of the implementation is provided in this chapter. The main blocks of real implementations are presented in Figure 9.8.

The user defines the Maximum Disconnection Interval (MDI), the training period, user profile etc, see Figure 9.8 step 1. The MDI is different for different types of applications such as VoIP, FTP etc. The user profile contains the raw Fuzzy rules. For example: working employee, travelling sales man, night shift employee. The profile can be omitted if there is no exact idea of the behaviour. The rules can only be created using the self-learning OP methods. The training period must be sufficiently large, so that all possible situations can be observed.

During the second step (Figure 9.8, step 2) the Fuzzy rules are loaded from the profile. The training data is collected during step 3 (Figure 9.8). The OP rules are created in step 4. The centres of the output membership functions are optimised according to the RLS method in step 5. The Fuzzy controller is ready for use in the LU procedure.

Let n be the number of past values considered in the Fuzzy controller. The last n Measured Intervals (MI) of the training data are used as an input for the newly designed Fuzzy controller, step 6 on Figure 9.8. The result of the Fuzzy controller - predicted Event Interval is a parameter of the transformation function. Additional parameters of the transformation function are set as described in 9.3.4. The input of the transformation function is the time point of monotone constant updates equal to the MDI. The result of the transformation function is the Update Time Point. During step 7, the Location Update procedure is executed. The result of the LU procedure is Boolean - true or false, dependent on whether an event (PoA change) has occurred within the interval or not. If an event has occurred then the middle of UI is the Measured Interval(MI). The prediction is repeated again, where the last n MIs are the input of the Fuzzy controller at step 6. If there is no event (PoA change) in step 7, then the next LU Update Interval is calculated and LU executed at the time point. Step 7 is repeated until the PoA change event occurs. The implementation runs infinitely and it interrupts when the user stops the execution.

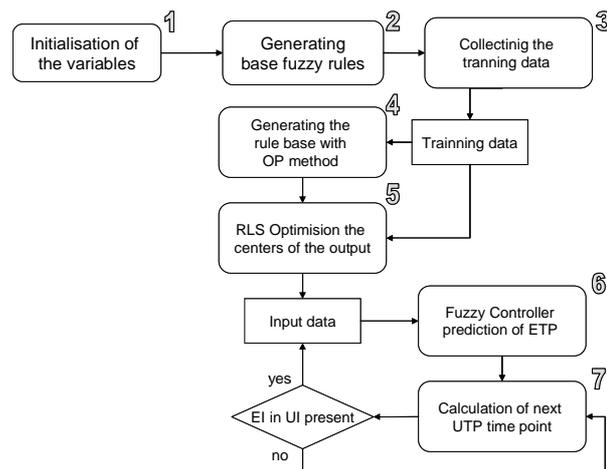


Figure 9.8: Implementation in real environment

It is possible that multiple PoA changes (Event Time Points) occur in one Update Interval(UI). The main problem is that this cannot be detected by the LU procedure. There is practically no difference between one ETP and multiple ETP in single UI. The procedure returns Boolean values: true if at least one ETP happened and false if not. It is not possible to detach multiple ETP in UI and this is not a problem limited to Fuzzy controllers but extends to all algorithms. Two or more events (PoA changes) in one LU interval lead to a wrong calculation of the MI and this introduces incorrect input values to the Fuzzy controller. If there are only a few UIs with two or more PoA changes, there is no observable impact of the long-term performance. Multiple PoA changes within the intervals in the training time are very difficult to handle and lead to incorrect rules bases. Incorrect training causes incorrect predictions. To recapitulate: multiple ETP in the training could have an impact on the performance and must be minimised. In the simulation experience, less than 3% double events do not cause measurable result problem.

9.4 Simulation of M-LU with adaptive Fuzzy Logic

A practical simulation of the Location Update procedure was carried out and it is described in this chapter. The simulation was made with Matlab 7.0 on a PC with 1.1 GHz and 512 RAM. The simulation is performed offline, so the PC speed aspect is irrelevant. The Matlab program is written with objectives to deliver precise results.

The simulation is programmed without involving the absolute time in any direct form. There is no indicator for the absolute time in the simulation. The Event Intervals are handled sequentially in loop (s. Figure 9.9). To handle dependencies on the absolute time, the method described in chapter 8.5.2 is used. The absolute time dependencies are circular since the time component is periodic. For example: there is a day cycle starting at 0h to 24h. These absolute time dependencies are solved with a simple pre-processing of the Event Intervals. The events are set relative to a point in the period (for example midnight of day).

The Fuzzy rules in the simulation are created through training with the One Pass method. There are no expert knowledge rules. The simulation is proof of concept of the suggested method. Involving expert knowledge can tamper the results. Poor or optimal expert rules significantly influence the performance. In this test, the target is to test the PDF approximation and OP method. In an extreme case, if there are optimal expert rules, then there is not need of other rules or any PDF.

The simulation is not carried out in the same way as the real time implementation, since only the precision of the method is of interest. For each Event Interval the necessary updates and their size is calculated. In contrast, in real implementation the updates are executed one by another until they reach the Event Time Point. The simulation consists of the following major blocks shown in Figure 9.9:

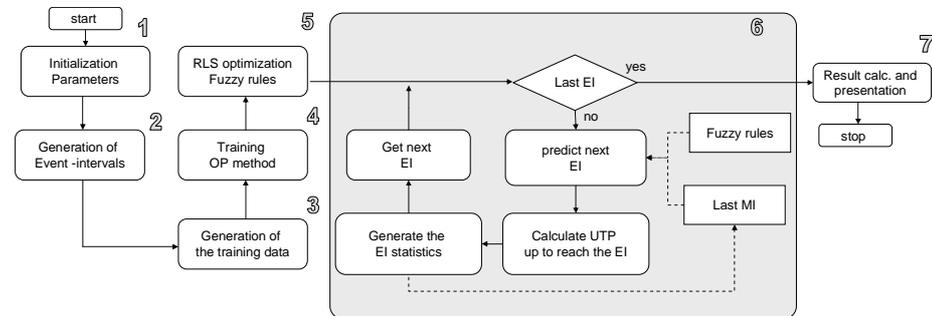


Figure 9.9: Simulation structure

- 1) *Initialisation.* The general parameters are defined, such as: number of Event Intervals, number of training samples, number of inputs, number of Fuzzy rules in the controller and maximum connection lost (MDI) tolerated by the application. The parameters are set manually.
- 2) *Generation of absolute Event Intervals.* Function generates semi random Absolute Event Intervals. The data varies from sinus function to real gathered data. The values are defined in 7.5.1.
- 3) *Calculation of the training samples.* The Update Intervals in the training are constant and equal to the MDI. The Measured Interval is in the middle of the Update Interval.
- 4) *Generation the Fuzzy rules.* The Fuzzy rules are calculated in the extra procedure named *op*. The function returns the centres of the antecedent/consequent membership bases. There is no need of the linguistic name of the membership functions. The centre of the membership function defines the rules fully because the membership intervals are predefined. All antecedent membership functions are triangular. The consequent functions are single tone.
- 5) *Optimization of the consequent membership rules.* The function - *training_rls* – implements the RLS to optimise the performance.
- 6) *Event interval procedure.* For each EI, the size of the UI is calculated. Furthermore, the number of UI needed to achieve this result is calculated (s. Figure 9.9). The variable remainder is calculated in every repetition. The remainder is used to align the starting point of the Event Interval to the filter cycle beginning as described in chapter 8.5.1. All results from the calculation are stored in an array for later analysis.
- 7) *Simulation result presentation.* The last procedure (*results_plot*) generates statistics and creates plots. This last procedure provides information on the quality of the Fuzzy controller. The statistics compare the given method to monotone constant intervals with the same number of updates. The result can be compared also to Particle Filter (chapter 8.6) and extended Kalman Filter (chapter 10.9). The monotone constant interval is currently the most implemented algorithm for this type of algorithm. The results analysis in the same way as described in chapter 7.5.2. Practically, the same Matlab procedure is used with the same reference variables.

9.5 Simulation results

The absolute Event Intervals are generated in the same way for all algorithms described in 7.5.1, thus the Particle filter (chapter 8) and Kalman Filter (chapter 10). The results are presented and analysis is described in 7.5.2.

The Fuzzy controller contains 100 rules. The 4 past values are considered in the prediction controller (input values). The Event Intervals are generated by the program code in the first 4 cases. There are 6000 intervals (samples) generated. The first 1000 of them are used for the training of the method, thus in the performance simulation is carried out on 5000 Event Intervals (samples of 1001 to 6000). The training is done with 1000 samples, which is 16.67% of all samples.

9.5.1 White noise

The Absolute Event Intervals (AEIs) are generated by white noise with some constant shifting as defined in chapter 8.6.1. The maximum Disconnection Interval is 5 sec. The membership interval is 75 sec. The membership interval is 15 times greater than the Maximum Disconnection Interval (MDI). In this way with fewer rules, more value combinations can be covered. The Event Intervals and the Measured Intervals (estimated EI) ones are shown in Figure 9.10. The results are summarised in the histogram Figure 9.11.

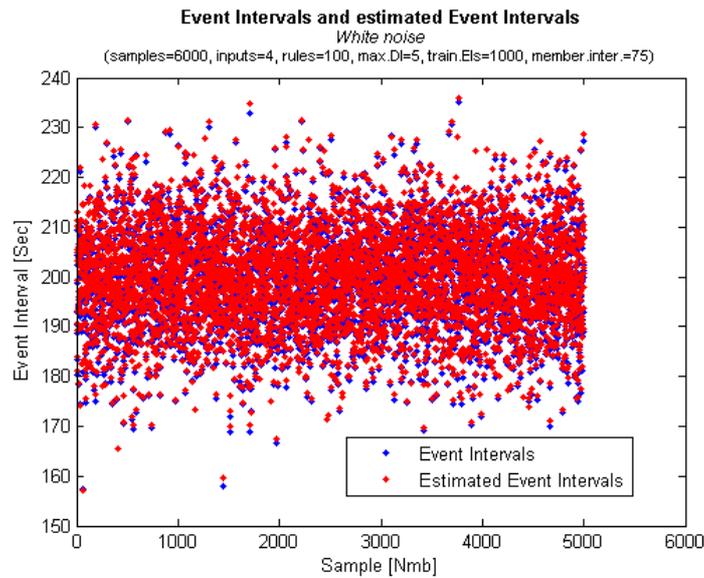


Figure 9.10: White noise, Event Intervals and posterior estimated Event Intervals

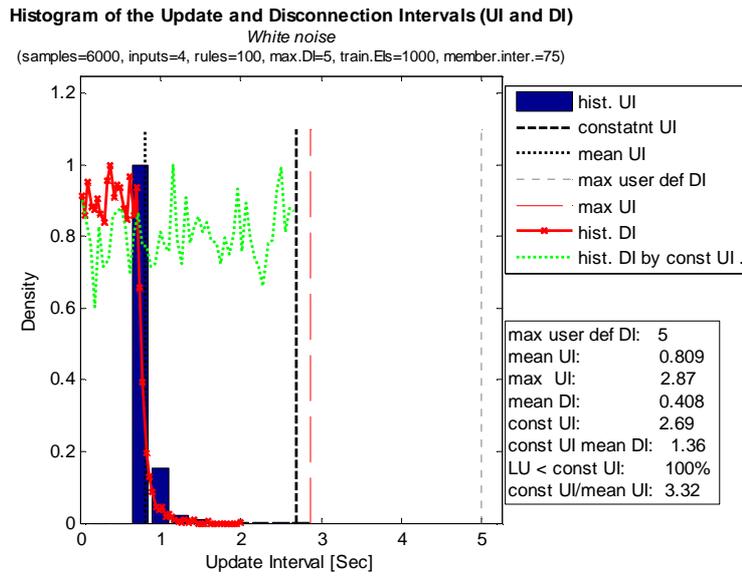


Figure 9.11: White noise, histogram results

The histogram (Figure 9.11) shows excellent results. In conjunction with the transformation function, the Fuzzy controller delivers a very good Update Interval distribution. All updates executed achieve better performance than the monotone constant Update Interval. To remind the readers, the constant Update Interval is calculated considering the same number of updates for the simulation time. The mean Disconnection Interval (mean DI) by constant Update Interval (const UI) is 3.32 times greater than the Fuzzy mean UI. There is a clear out performance of the Fuzzy method over the Particle filter method for the white noise values (see chapter 8.6.1). The result is an effect of the transformation function. The transformation function has the same form as the PDF of white noise, see 9.3.4. The Fuzzy controller plays no major role in this case, since the only prediction is the mean value. The results of the simulation are impressive. The Update Intervals for Event Intervals are shown in Figure 9.12. This can be considered as best case.

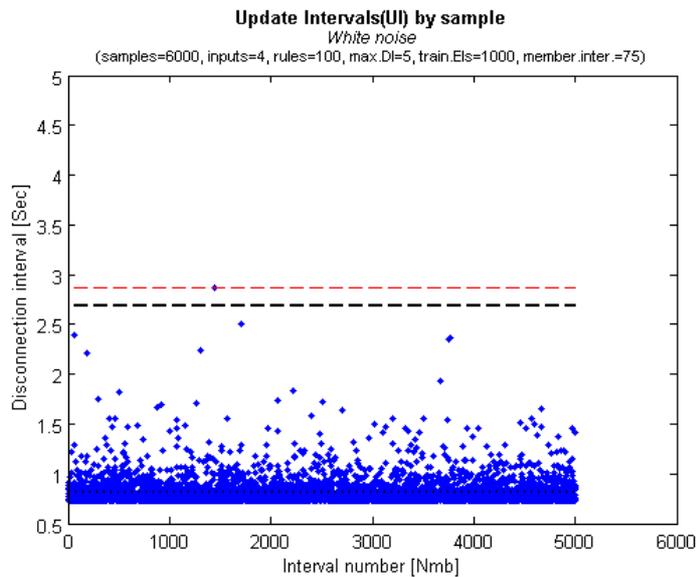


Figure 9.12: White noise, update intervals with Event Time Point

9.5.2 Two rotating white noise sources

In this experiment, the Event Time Points are generated using two randomly rotating white noise sources. The equation is the same as that described in chapter 8.6.2. The Maximum Disconnection Interval, membership interval etc are the same as in previous simulation run in chapter 9.5.1.

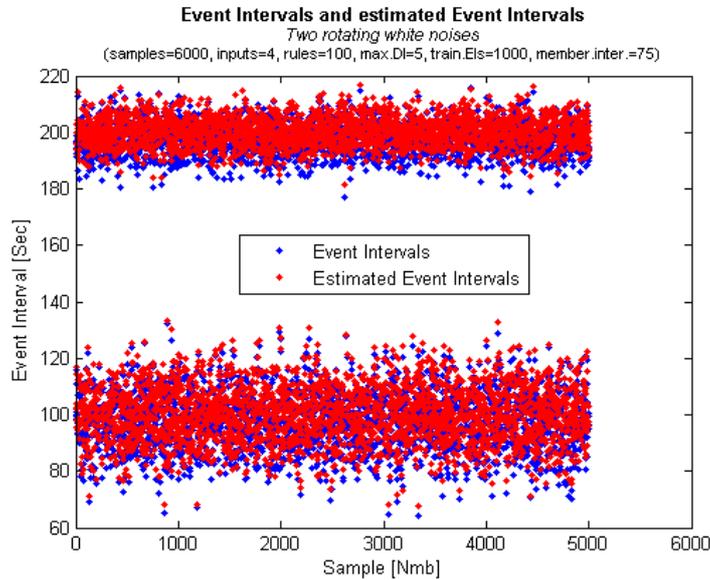


Figure 9.13: Two rotating white noises, Event Intervals and estimated EI

The Event Intervals can be seen in Figure 9.13. The PDF of the Event Interval was already shown in chapter 8.6.2.

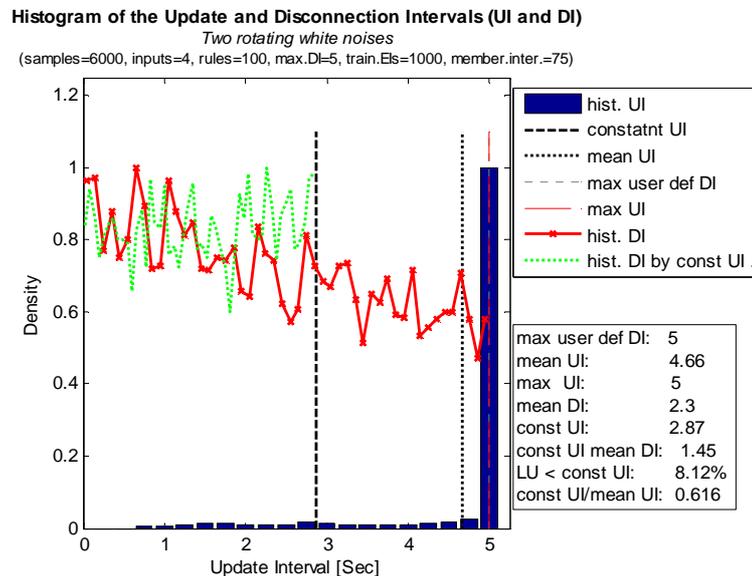


Figure 9.14: Two rotating white noises, result histogram

The resulting histogram shows that the Fuzzy algorithm is overstrained with this task. The algorithm cannot detach the data's nature in the training period. Only 8.12% of the updates are better than the referenced constant updates. The mean constant Update Interval (const UI) is smaller than the mean Update Interval with ETP (mean UI) by a factor of 0.616. The results show the poor performance of the Fuzzy controller. It cannot handle the rotating signals. Examining the Figure 9.15 (Update Interval by sample) shows that the problem is not

concentrated on certain regions of the updates. The algorithm systematically fails to predict the signal.

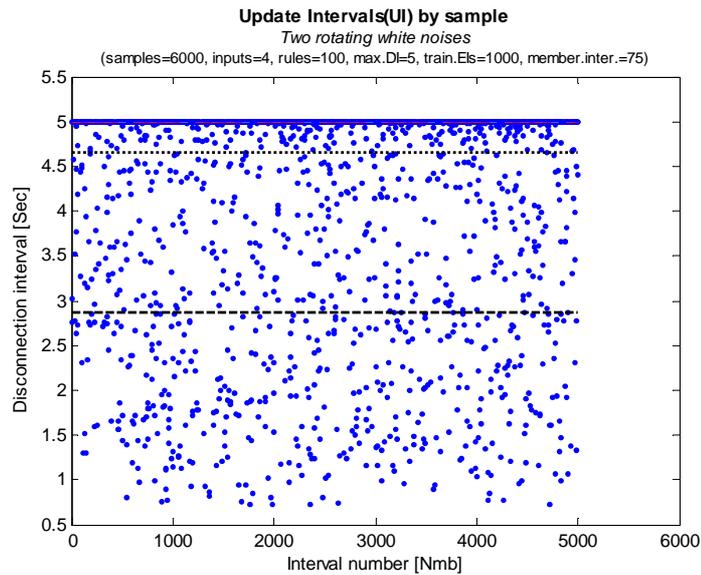


Figure 9.15: Two rotating white noises, update intervals with ETP

The reason for this underperformance is the random rotation of the two signals. This overlay random behaviour cannot be captured by the training of the Fuzzy controller. The transformation function cannot compensate for this failure in the prediction as in the previous case. The method cannot be recommended for this type of moving pattern.

9.5.3 Sinus based EI with white noise

In this simulation, a sinus based signal with white noise is used, as described in chapter 8.6.3 (σ is set to 5). The idea of this simulation run is to show the importance of choosing the right parameter. The simulation is made with a membership function interval of 15 sec, which is quite little. In the previous two experiments, the membership interval was 75 sec.

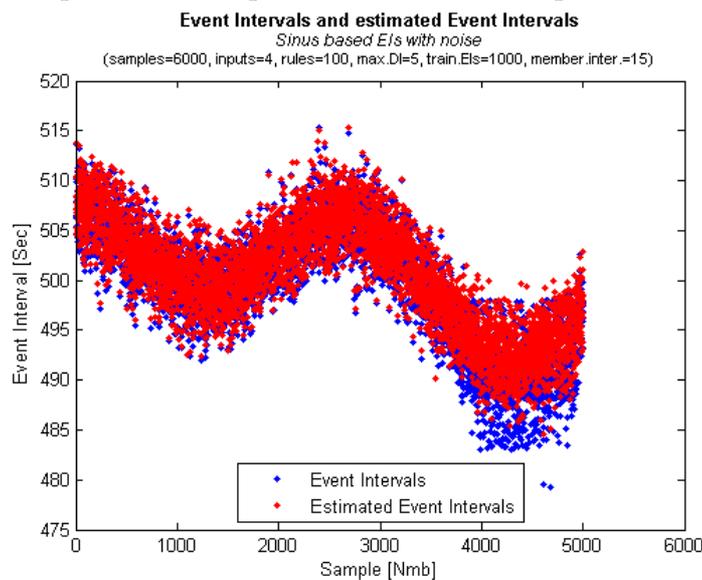


Figure 9.16: Sinus base EI with white noise, event time intervals and estimated EI

The Event Intervals (EI) and the posterior estimated EI are shown in Figure 9.16. The typical sinus-like form is easy to see. After sample 3800 there is quite a big difference between the real value and the estimated ones. Obviously, the algorithm fails to predict the values in this region. This can also be verified by examining Figure 9.17, where the Update Intervals are shown with ETP. The intervals are at the Maximum Disconnection Interval after the sample of 3800, thus 5 sec. The algorithm has falls back to the maximum values in this range.

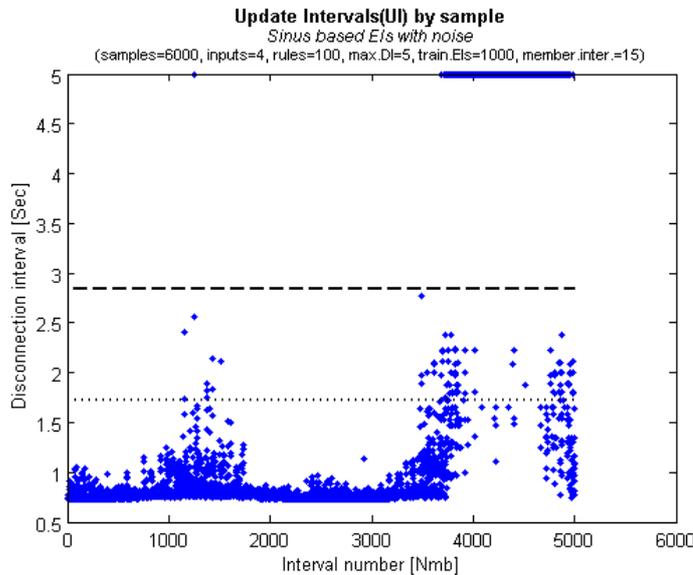


Figure 9.17: Sinus based EI with white noise, update intervals with ETP

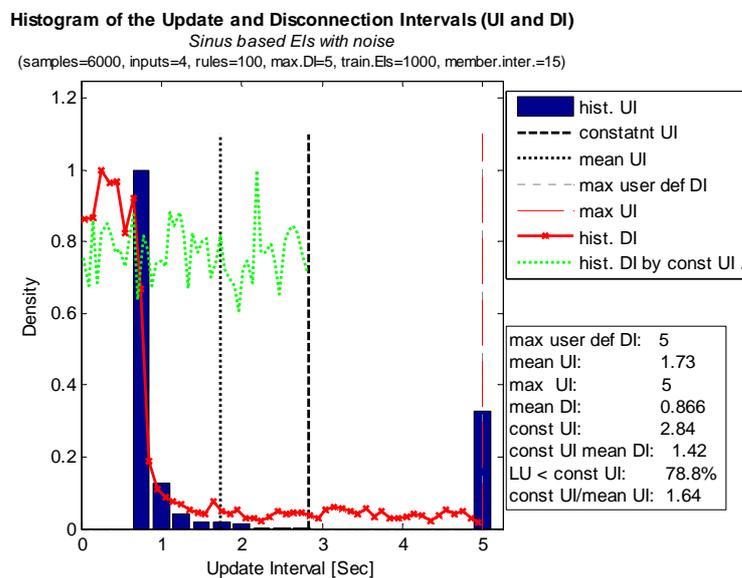


Figure 9.18: Sinus based EI with white noise, result histogram

The algorithm fails to predict in this region because of the insufficient training time. The training has not included this type of input-output values and consequently no rules were created. The even intervals are smaller than the previous ones and the training did not include them. Here, the issue of too short training is manifested. It can be concluded that the training must contain all possible combinations of input-output in order for the method to deliver good results. This can be partially compensated by choosing a sufficiently large membership

interval. The interval was set smaller than was optimum in the simulation, to demonstrate the effect.

The histogram is shown in Figure 9.18. The results are very good in general, although there is a problem in the last region. The Update Intervals are small (Figure 9.17). 78.8% of the updates were better than the constant update and the Update Interval is 1.64 times smaller than the constant Update Interval.

Repeated simulation with a membership interval of 75 significantly improves the performance. More than 90% of the update with ETP is smaller than the constant update. The results are similar to the first case with white noise. The method can excellently predict this type of signal when the training is sufficient.

9.5.4 Non linear EI with white noise

The Absolute Event Intervals are calculated with recursive non-linear equations with white noise defined in chapter 8.6.4. The membership interval is 75 sec.

The results are presented in Figure 9.20. The method has a 91.1% advantage in comparison to the constant update by sample. The mean update value is smaller than the constant Update Interval by a factor of 2.04. In this difficult case, there is a significant improvement in performance over the constant intervals. The performance is even better than the method suggested by Particle filter in chapter 8.6.4. The Update Interval histogram has the typical exponential distribution, having a high bin at small Update Intervals. The prediction was excellent in the majority of cycles. The good performance can be explained by the recursive form of the Fuzzy system, which is the same as the generated data. The Event Intervals are shown in Figure 9.19. The Update Intervals with ETP are in Figure 9.21.

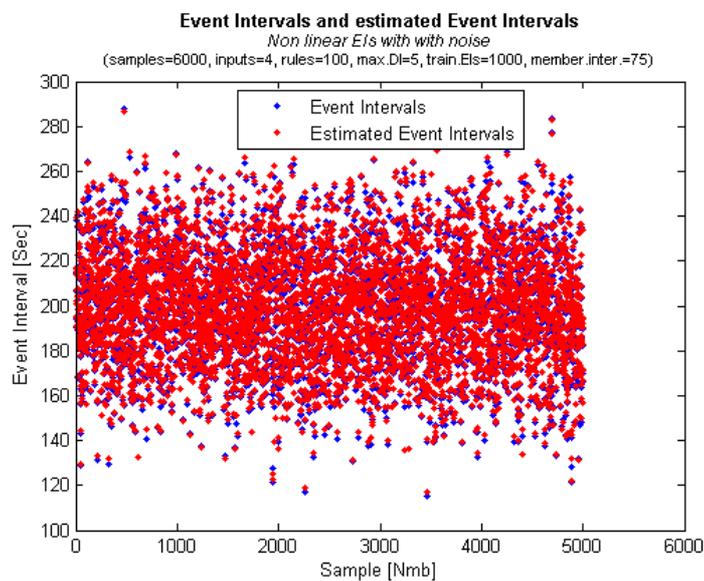


Figure 9.19: Non linear recursive EI, Event Intervals and estimated EI

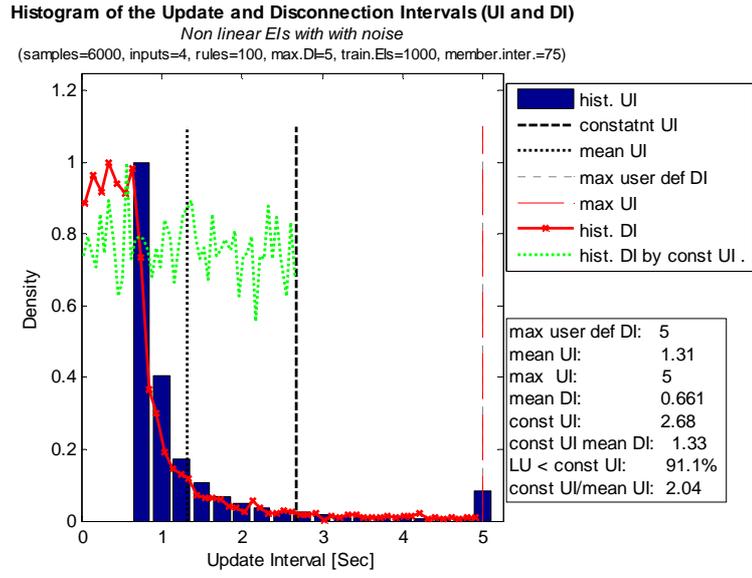


Figure 9.20: Non linear recursive, results histogram

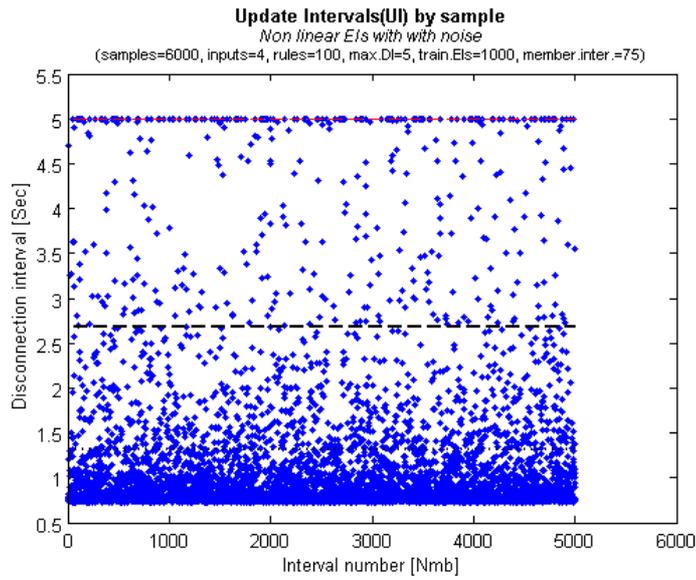


Figure 9.21: Non linear recursive, update intervals with ETP

9.5.5 Real data

Real data gathered from operation networks is used in this simulation, see 7.5.1. The Maximum Disconnection Interval is 300 sec. There are 2500 gathered Event Intervals within 5 days. The first 1000 are used for training (2.5 days). The membership interval is 4500 sec, which aids improved interpolation of the states. There is a day cycle in the Event Time Points. The Event Intervals are pre-processed set relative to midnight. In this way, the algorithm can handle cycle events, see chapter 8.6.5.

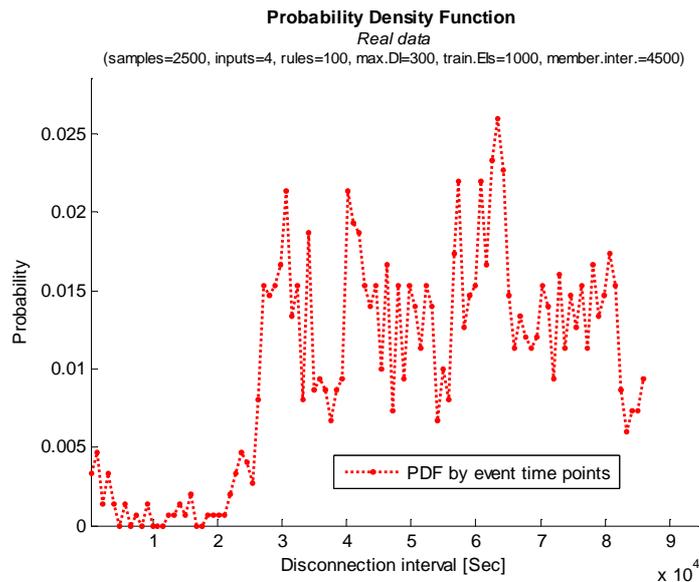


Figure 9.22: Real data, PDF of Event Intervals

The PDF is presented in Figure 9.22. It is easy to notice the day cycle with high activity during the day and less activity in the night.

The results are presented in Figure 9.23. The new method performs twice as well as the constant update. The mean UI of the new Fuzzy method is smaller than the constant update by a factor of 2.05. More than 92% of all updates are smaller than the constant updates. It can be concluded that the method shows impressive performance in the real time data. The method has much better results than the Particle filter on the same data. Again, there is the typical Update Intervals histogram (blue bins) with definite concentration on the small values. There are no blackouts (failures) of the algorithm, which means that the training parting terms of the real data was sufficient to create the necessary rules. The Update Intervals with sample are presented in Figure 9.24. The method can be recommended for this type of data.

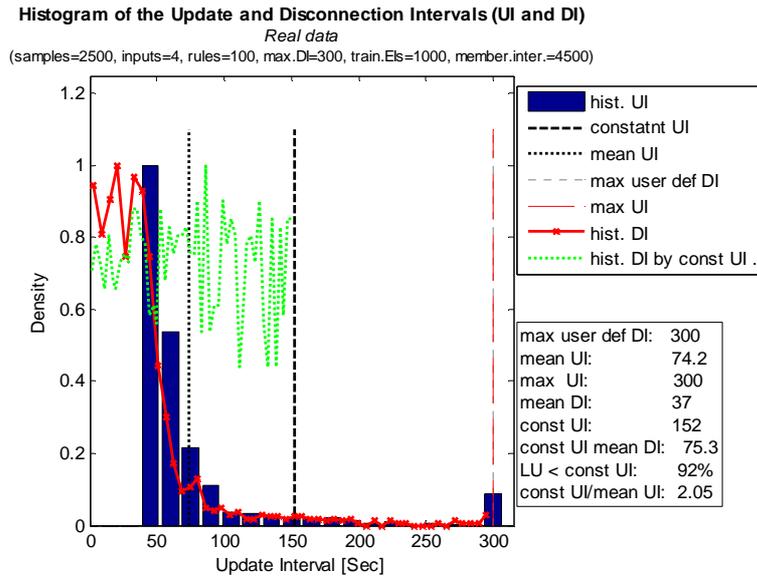


Figure 9.23: Real data, result histogram

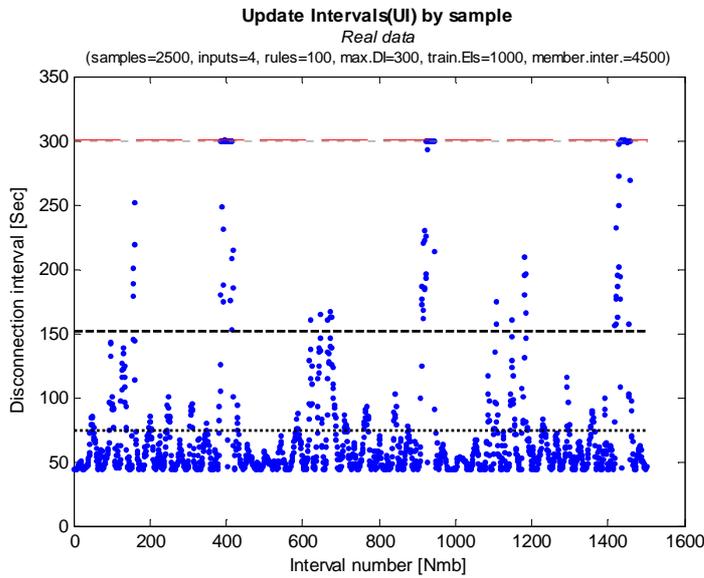


Figure 9.24: Real data, Update Intervals with ETP

9.6 Conclusion and future work

The simulation has verified that the new method delivers very good results. The method has shown excellent performance on the real data, even better than Particle filter (chapter 8). The method achieves very good results with sinus based and recursive data. An important requirement is to have sufficient training samples. Otherwise, poor result can be expected. Fewer training samples can be partially compensated by increasing the membership interval and interpolating in this way.

A major failure of the algorithm occurred when working with two randomly rotating white noise sources (see 9.5.2). The algorithm did not manage to create an adequate Fuzzy rule base. The reason was the large interval between the mean values of the noises, the small standard deviation and the random rotation of the generator. Better results could be achieved by drastically increasing the number of Fuzzy rules.

There is future work for improving the algorithm. It is out of the scope in this thesis since it is a more classical optimisation engineering problem. The main points can be summarised as:

- The Fuzzy controller parameters are set manually for obtaining good results. The parameters were chosen through making numerous simulations. These are selected systematically. The parameters should be fit automatically for practical implementation. It must be stressed that a bad parameter decreases the performance significantly even under the constant Update Interval. This is an optimisation task for the control engineering.
- Training should be implemented during the operation. The training period is limited to the interval before the normal operation of the controller. The method can be extended, so that the rule base is updated after every update/prediction cycle. The algorithm can handle a dynamically changing PDF function in this way. Furthermore, the risk of missing rules decreases, because of a steady feedback of the rule base (see 9.5.3). The implementation must regenerate the Fuzzy rules after every value received.
- Overlaying of multiple periodical dependencies should be implemented, as described in 8.7

9.7 References in chapter 9

- [1] Mendel, J.M., "Fuzzy Logic Systems for Engineering: a Tutorial," IEEE Proc., Vol. 83, pp. 345-377, March 1995.
- [2] Mouzouris, G., "Designing fuzzy logic systems," (G. Mouzouris, co-author), invited paper, IEEE Trans. on Circuits and Systems, Part II, vol. 44, pp. 885-895, Nov. 1997.
- [3] Klement, Erich Peter, Mesiar, Radko and Pap, Endre, "Triangular Norms"; Springer; July 1, 2000
- [4] Wang, L. et al. "Fuzzy Basis Functions, Universal Approximation, and Orthogonal Least Squares Learning," () IEEE Trans. on Neural Networks, vol. 3, pp. 807-814, Sept. 1992.
- [5] Wang, L. X. and Mendel, J., "Generating fuzzy rules by learning from examples," IEEE Transactions on Systems, Man, and Cybernetics, vol. 22, July 1992.
- [6] Haykin, Simon, "Adaptive Filter Theory", Prentice Hall, 2002.
- [7] Gordon, N., Salmond, D., and Smith, A. F. M., "Novel approach to nonlinear and non-Gaussian Bayesian state estimation", Proc. Inst. Elect. Eng., F, vol. 140, pp. 107-113, 1993.
- [8] Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T., "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking", IEEE, VOL. 50, NO. 2, 2002
- [9] Tzvetkov, Vesselin, "Optimization of update intervals in Dead-Peer-Detection using adaptive Fuzzy Logic". IEEE AINA, 2007
- [10] Tzvetkov, Vesselin, "Fast detection of disconnection using adaptive Fuzzy Logic", IEEE ICNSC, 2007

10 Mobile Location Update protocol based on extended Kalman filter

An alternative approach for M-LU is developed in this chapter, which is based on extended Kalman filter [3]. The filter is used to predict the coming Event Interval (EI) and its variance. A model for the distribution and measurement of EI is developed in order to use the extended Kalman filter. The Update Time Points are generated with a transformation function. This function converts constant update intervals into non-linear inverse proportional to the PDF of EI, as defined in chapter 7. The function has a parameter the predicted EI and the Maximum Disconnection Interval (MDI).

The text is organised as follows: The model definition is given in 10.2. The distribution of UTPs and the ideal transformation function are derived in 10.3. A suboptimal transformation function is created in 10.5 and 10.6. The extended Kalman filter is derived in 10.7. The simulation structure and results are presented in 10.8 and 10.9.

10.1 Contributions

The author contributions in this text are: (1) First, creating a model for Mobile Node movement on which Kalman filter can be applied, see 10.2. (2) Second, derivation of the extended Kalman Filter for the prediction of EI with semi standard deviation, see 10.7. (3) Third, creation of a novel transformation functions for the calculation of UTP according the predicted EI. Practical contribution is the simulation of the method and analysing the results in section 10.8.

The author has published the method and the simulation results in relation to SIP protocol in [7] at IEEE.

10.2 Model for extended Kalman Filter

At least two models are required for the description of dynamic system with Kalman filter. The first one describes the time evaluation of variable of interest. It expresses the natural properties of the process, for example the velocity on a free-falling body. The second describes the measurement. The measurement typically returns indirect the variable of interest, thus with uncertainty and noise. The equation gives the relation between the variable of interest and the result of the measurement. The variable of interest is also called unobserved (hidden) variable.

The system equations are commonly described with a state-space model [3]. The model works in a time domain and is very suitable for Multi Input Multi Output systems (MIMO). State-space notation uses differential equations for which the mathematical theory is very well developed.

Major requirement for deploying Kalman filter [2] is that the system must be linear and Gaussian. The extended Kalman filter works with non-linear system. It achieves linearization of the signal with truncated Taylor series and then uses the Kalman filter. Still the system must be Gaussian to be used with extended Kalman filter. The model equations must contain only Gaussian (Normal) noise component [6, 3]. The created two models for mobile Node movement must obey this requirement.

10.2.1 Natural properties of the Event Intervals

The natural properties of the Event Intervals (EIs) distribution in time are unknown, thus movements of the Mobile Node cannot be defined at this stage. Unfortunately, the Kalman

filter cannot be implemented without the evaluation model as already mentioned. An equation must be defined which has Gaussian distribution on the one hand and on the other hand, does not restricts the potential behaviour of the Mobile Node.

A random walk is assumed for the EI time evaluation. Let x_k be the Event Interval at step k . The random walk is defined by $x_k = Ax_{k-1} + w$, where $w = N(0, \sigma)$ is white noise and vector A called transformation matrix (Vector is matrix with one dimension equal to 1). Later on, the vector A is set to 1 (one) in the simulation.

The standard deviation of the noise component should not restrict the values of the EI. For this reason, the standard deviation is calculated recursively from the past Measured Intervals at every filter cycle.

10.2.2 Measurement model

The second model represents the measurement of the Event Interval (EI), denoted as Measurement Interval. The exact measurement of the EI is not possible. The EI can only be narrowed down to the Update Interval (UI) with the EI but not to an exact numerical value. The Kalman filter requires numerical values, so some assumption must be made.

The assumed Measured Interval (MI) is in middle of the Update Interval (UI) with EI. The absolute error of MI is minimized at this point. Let y_k be the Measured Interval of the k^{th} prediction/update cycle, thus: $y_k = f(x_k)$. The x_k is the EI of the k^{th} cycle. The measurement function $f()$ is returning the MI of the EI

The MI depends on the UI. The UI depends on the predicted (prior estimated) EI, since the Update Time Points (UTP) are calculated with transformation function, see 10.4. The predicted EI is denoted as \hat{x}_k^- , see 10.7. Let $T()$ be the transformation function and x_{\max} be the MDI. The measurement function is then:

$$y_k = f(x_k) = \frac{1}{2} \left(T(x_{\max} \cdot T^{-1}(x_k - \hat{x}_k^-) \bmod x_{\max}) + T((x_{\max} + 1) \cdot T^{-1}(x_k - \hat{x}_k^-) \bmod x_{\max}) \right)$$

The extended Kalman filter requires the standard deviation of the measurement. Without this definition it is impossible to derivate the filter. Unfortunately, the standard deviation of measurement function cannot be found analytically, since the function is interrupted.

Numerical value of standard deviation is not directly important for the M-LU method. The value in M-LU must be proportional to some degree to the standard deviation. The standard deviation in sense of the Kalman filter is small for values with high probability and large for values with low probability. A variable called semi standard deviation can replay the standard deviation when it has the same characteristic. In this way, the extended Kalman filter will deliver correct prediction values but numerically wrong standard deviation. The resulting semi standard deviation will be proportional to some degree, which is sufficient of M-LU. The semi standard deviation can be Update Interval with event, which is defined by:

$$v_k = T((x_{\max} + 1) \cdot T^{-1}(x_k - \hat{x}_k^-) \bmod x_{\max}) - T(x_{\max} \cdot T^{-1}(x_k - \hat{x}_k^-) \bmod x_{\max}),$$

where v_k denotes the semi standard deviation of the k the cycle.

10.3 Distribution of the Update Time Points

The extended Kalman filter delivers single value of the predicted EI and the semi standard deviation. These are not sufficient to construct the Probability Density Function (PDF). The PDF is necessary because the UIs must be set inversely proportional to the PDF, as required in chapter 7. For this reason, some assumption must be made.

The PDF of EI has Normal distribution with maximum at the predicted EI. The standard deviation of the Normal distribution is set proportional to the result deviation. The proportion is defined later on by the derivation of the transformation function.

10.4 Transformation function

As already discussed in chapter 7, the length of the UI must be inversely proportional to the PDF of EI. Low values of PDF define large Update Intervals and high PDF the Update Interval must be small. An abstract curve of PDF (Normal distributed) and Update Interval is shown in Figure 10.1 (and previously shown in Figure 9.6).

The Update Intervals must not exceed the Maximal Disconnection Interval (MDI) as additional requirement in chapter 7. Together with the condition for inversely proportional to Normal condition, the Update Intervals can be defined generally by:

$$L(t) = a - b \cdot N(t, \mu, \sigma), \quad \text{Equation 10.1}$$

where $L(t)$ is the length of the Update Interval at time t . The $N(t, \mu, \sigma)$ denotes the Normal distribution of EI at point t with parameter mean μ and standard deviation σ . The constant a denotes the Maximal Disconnection Interval (MDI) defined by the application. Variable b defines the pitch of the bell curve. The popular normal distribution is defined by:

$$N(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

The Equation 10.1 is the general form of the Update Interval length, so that it satisfies the requirement of chapter 7. Next, the coefficient values must be precisely calculated in order to deploy it.

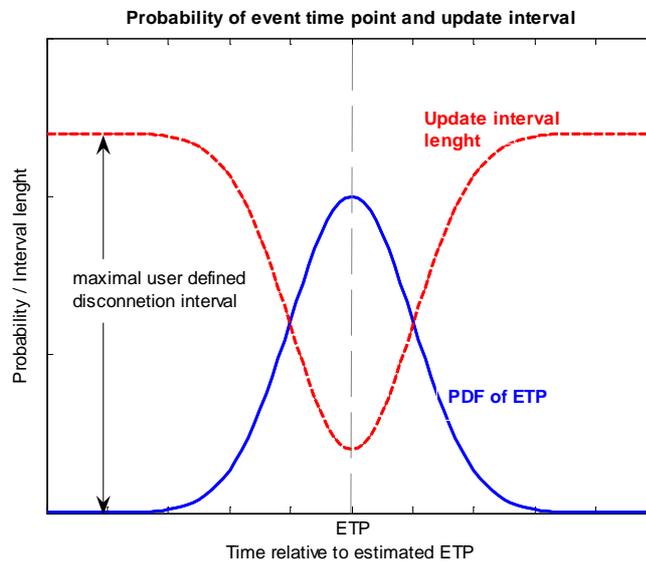


Figure 10.1: Probability of Event Interval distribution and length of Update Interval length

The key idea in this work is to derived transformation function, which converts constant intervals in ones satisfying Equation 10.1. This is very suitable for practical implementation. The input/output values are easy to calculate. The transformation works as black box from the perspective of the application.

The transformation function has as input the time point of the constant Update Interval. The output is the Update Time Point considering the PDF of EI and the Maximal

Disconnection Interval (MDI). Let $T()$ be transformation function. Then $T(x_{i+1}) - T(x_i) = L(x_i)$, where x_i, x_{i+1}, \dots are the constant update time point. This key equation is transformed as follows:

$$T(x_{i+1}) - T(x_i) = L(x_i) \quad \text{Let } x_{i+1} - x_i \rightarrow 0$$

$$\frac{\partial T(x)}{\partial x} = L(x) = a - b \cdot N(x, \mu, \sigma)_i$$

$$T(x) = \int a - b \cdot N(x_i, \mu, \sigma) dx$$

$$T(x) = \int a - b \cdot \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) dx$$

$$T(x) = ax - \frac{b\sqrt{2}}{\sigma\sqrt{\pi}} \operatorname{erf}\left(\frac{x}{\sigma} - \frac{x}{\mu}\right) + C,$$

where

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

is called error function. C is constant. The coefficients of $T()$ can be generalised for simplification to:

$$T(x) = ax - u \cdot \operatorname{erf}\left(\frac{x}{k}\right) + C \quad \text{Equation 10.2}$$

where a, u, k and C are new constants.

This is the definition of the ideal transformation function, so that linear update intervals are transferred into non-linear ones. The function satisfies fully the Equation 10.1, which is inversely proportional to the PDF of EI and does not exceed the MDI. An abstract plot of the function is shown in Figure 10.2. The constant update intervals are at the ordinate (y-axis). The result at abscise (x-axis) shows the non-linear Update Intervals. The intervals become shorter near the predicted Event Interval (zero in the figure) and increase infinitely to the MDI. The predicted EI is set to 0 at the diagram and all other points relate to it.

Important properties of the function are at the extremes. In infinite ($\pm \infty$), the second term of the function goes to zero:

$$\lim_{x \rightarrow \pm\infty} u \cdot \operatorname{erf}\left(\frac{x}{k}\right) = u \lim_{x \rightarrow \pm\infty} \int e^{-x^2} dx = u \int \lim_{x \rightarrow \pm\infty} e^{-x^2} dx = 0,$$

since

$$\lim_{x \rightarrow \pm\infty} e^{-x^2} = 0$$

The transformation function at infinity is:

$$\lim_{x \rightarrow \pm\infty} T(x) = \lim_{x \rightarrow \pm\infty} (ax - u \cdot \operatorname{erf}\left(\frac{x}{k}\right) + C) = ax + C$$

The transformation function becomes linear with coefficient a and shift C . The linearity means constant interval. The Update Interval is reaching the defined maximum and it is not

growing any further, thus the Maximum Disconnection Interval (MDI). The linear intervals are transferred into smaller intervals for values near to the predicted EI (zero point) because the probability of EI increases there.

The coefficients of $T()$ are calculated in the following section 10.5

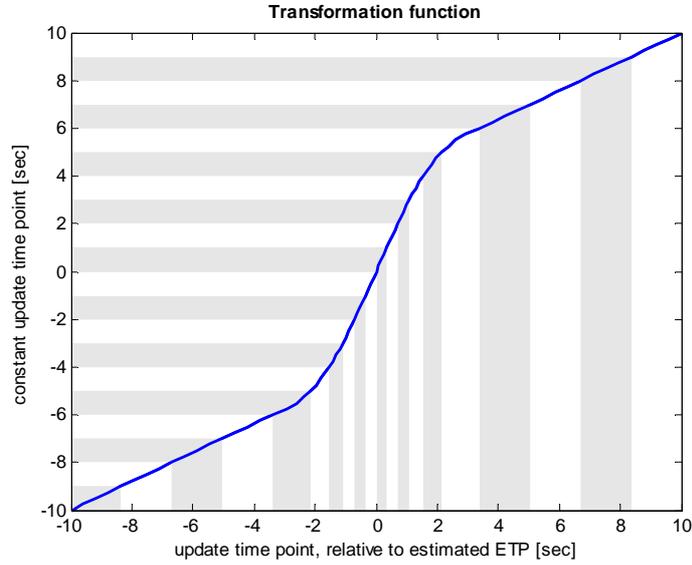


Figure 10.2: Transformation function example

10.5 Approximated Transformation function

Equation 10.2 has only theoretical character. Unfortunately, it cannot be implemented since the straight and inverse transformation does not exist. The function $T(x) = ax - u \cdot \text{erf}\left(\frac{x}{k}\right) + C$ is a complementary function and its inverse $T^{-1}(x)$ cannot be found analytically.

The idea is to approximate the $T()$ by a function that has an inverse. The function is approximated to the sum of functions in different intervals. Analysis of Equation 10.2 shows that the term $u \cdot \text{erf}\left(\frac{x}{k}\right)$ dominates for small x values and the term ax dominates for big x . The transformation function can be approximated to these terms in different intervals. The approximation is:

$$T'(x) = \begin{cases} u \cdot \text{erf}(x/k) + C_1 & x \in [-l, +l] \\ ax + C_2 & x \in (-\infty, -l) \cup (l, \infty) \end{cases}$$

Then the inverse function is:

$$T'^{-1}(x) = \begin{cases} k \cdot \text{erf}^{-1}((x - C_1)/u) & x \in [T'(-l), T'(l)] \\ (x - C_2)/a & x \in (-\infty, T'(-l)) \cup (T'(l), \infty) \end{cases}$$

Splitting the function in this way leads to a sub optimal solution but it can be implemented. This approximation can be done in other ways but the author considers this one to be the most convenient one because of its simplicity.

10.6 Coefficients of the transformation function

The coefficients of the approximated function must be derived in order to use the function. Conditions must be defined so that coefficients can be calculated from them. There

are not sufficient objective conditions and therefore subjective ones are used additionally. They are listed here:

1) The first condition is that the transformation function and its inverse must be continuous (not interrupted). Otherwise, it is not possible to find the inverse for some values. This gives a relation between the two functions at the crossover point l :

$$u.erf(l/k) + C_1 = a.l + C_2$$

The functions must have the same value at this point.

2) The Update Interval at the most probable point, predicted EI, must tend to zero. This means that $C_1 = 0$.

3) The cross point l is defined using subjective knowledge. The cross over point defines the region $(-l, l)$ where the intervals become inverse proportional to the probability. Updates are constantly outside this interval since the transformation function is linear there. The predicted Event Interval is reasonable to be equal to l . Let $l = P_k^+$, where P_k^+ is the predicted Event Interval. The k index denotes the prediction cycle. The plus sign stress that it is the predicted value – prior estimation. In this way, the Update Intervals decrease until reaching the estimated ETP and then increase until reaching one constant value.

4) The linear update interval is the Maximum Disconnection Interval defined by the application. It is the input of the transformation function. The value is chosen to simplify the calculation. Any other constant can be used and this will lead to the same performance.

5) The output of the transformation function (the non-linear Update Interval) must also not exceed maximum Disconnection Interval. Following from condition 4), the derivative of $T(x)$ must be greater than 1 in $x_k \in (-P_k^+, P_k^+)$:

$$\frac{\partial}{\partial x} u.erf(x_k/k) \geq 1 \quad x_k \in (-P_k^+, P_k^+)$$

The linear function must be equal to MDI outside the interval $x_k \notin (-P_k^+, P_k^+)$. The result must be equal to 1 outside this interval:

$$\frac{\partial(ax + C_2)}{\partial x} = 1 \Rightarrow a = 1$$

6) Using the subjective observation of $erf()$, it can be seen that the $erf()$ reaches its maximum at infinity. On the other hand, the function reaches 95% of its maximum quite rapidly at approx. $x=1.38k$, see Equation 10.2. The last 5% of the increase can be approximated to an almost horizontal line. The linear part of the function is not relevant to our transformation. The condition is defined with this empirical and subjective consideration: The $erf()$ function reaches 95% of its maximum at the cross point between the components functions. The conditions 1) and 2) give:

$$u.erf(P_k^+/k) = P_k^+ + C_2$$

The result by setting the $erf()$ is:

$$erf(P_k^+/k) = 0.95 \Rightarrow P_k^+ + C_2 = 0.95u$$

7) The transformation function has its minimum at the predicted EI, where the probability is highest. The predicted ETP is set to zero and all points in relation to it. The transformation function is an odd function returning positive values. This condition facilitates the practical implementation and does not change the performance.

These seven conditions deliver the solution for the coefficients of the transformation function. The following system equations are obtained by setting all of the conditions together:

$$\begin{cases} \frac{\partial}{\partial x} u \cdot \text{erf}\left(\frac{x_k}{k}\right) \geq 1, x_k \in [-P_k^+, P_k^+] \\ u \cdot \text{erf}\left(\frac{P_k^+}{k}\right) = P_k^+ + C_2 \\ P_k^+ + C_2 = 0.95u \end{cases} \quad \text{Equation 10.3}$$

The first equation can be transformed to:

$$\begin{aligned} \frac{\partial}{\partial x} u \cdot \text{erf}\left(\frac{x_k}{k}\right) &= \frac{\partial}{\partial x} u \frac{2}{\sqrt{\pi}} \int_0^{\frac{x_k}{k}} e^{-t^2} dt \\ u \frac{2}{k\sqrt{\pi}} e^{-(x_k/k)^2} &\geq 1, x_k \in [-P_k^+, P_k^+] \end{aligned}$$

The exponential function has its minimum at the edge of the interval $[-P_k^+, P_k^+]$. The condition will be always true if the exponent of the function is set to 1 for the minimum. This is a partial solution of the equation and can be written as:

$$\begin{aligned} u \frac{2}{k\sqrt{\pi}} e^{-(P_k^+/k)^2} &= 1 \\ \ln\left(\frac{2u}{k\sqrt{\pi}}\right) - \left(\frac{P_k^+}{k}\right)^2 &= 0 \end{aligned}$$

Following, the partial solution of system in Equation 10.3 is:

$$\begin{cases} \ln\left(\frac{2u}{k\sqrt{\pi}}\right) - \left(\frac{P_k^+}{k}\right)^2 = 0 \\ u \cdot \text{erf}\left(\frac{P_k^+}{k}\right) = P_k^+ + C_2 \\ P_k^+ + C_2 = 0.95u \end{cases}$$

This system delivers a single solution of Equation 10.3 and not necessarily all solutions. One solution is sufficient for M-LU calculations.

The new system has three variables and three equations, thus it can be solved. Its results are:

$$k = \frac{P_k^+}{\text{erf}^{-1}(0.95)}, \quad u = \frac{k\sqrt{\pi}}{2} e^{2\text{erf}^{-1}(0.95)^2}, \quad C_2 = 0.95 \cdot \frac{\sqrt{\pi}}{2} e^{\text{erf}^{-1}(0.95)^2} - P_k^+$$

To recapitulate, the transformation function is:

$$T'(x) = \begin{cases} u \cdot \text{erf}(x/k) & x \in [-P_k^+, +P_k^+] \\ x + C_2 & x \in (-\infty, -P_k^+) \cup (P_k^+, \infty) \end{cases}$$

Then the inverse function is:

$$T'^{-1}(x) = \begin{cases} k \cdot \text{erf}^{-1}(x/u) & x \in [T'(-P_k^+), T'(P_k^+)] \\ x - C_2 & x \in (-\infty, T'(-P_k^+)) \cup (T'(P_k^+), \infty) \end{cases}$$

where

$$k = \frac{P_k^+}{\text{erf}^{-1}(0.95)}, \quad u = \frac{k\sqrt{\pi}}{2} e^{2\text{erf}^{-1}(0.95)^2},$$

$$C_2 = 0.95 \cdot \frac{\sqrt{\pi}}{2} e^{\text{erf}^{-1}(0.95)^2} - P_k^+$$

An example of the approximated function with a linear Update Interval of 1 and prior estimation of 2 is shown in Figure 10.3. The zero point at the abscise is a predicted Event Time Point (ETP). The time values at abscise are relative to ETP, see condition 5). The function obeys the required conditions: it becomes linear at infinite and is not linear for values near to the ETP. The intervals become smaller upon nearing the zero point, thus the probability becomes greater – the intervals smaller.

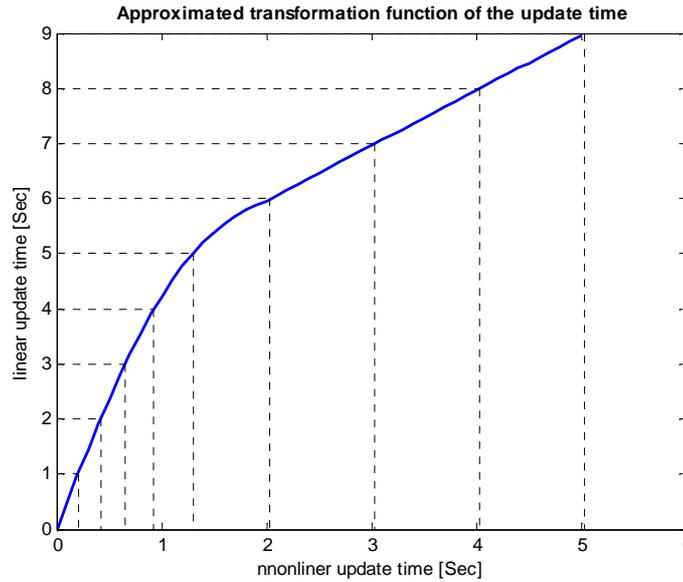


Figure 10.3: Transformation function example

10.7 Extended Kalman filter for M-LU

There are two phases in the filter model: update and prediction. They are executed sequentially at every filter cycle. The cycle is denoted at the sub index of each variable. The Bayesian inference (see 8.2.2) expresses the optimal dependencies. The equations are briefly repeated here. Let the posterior PDF function at $k-1$ th filter cycle be $p(x_{k-1}|y_{1:k-1})$. The prior PDF is calculated in the prediction phase using:

$$p(x_k|y_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|y_{1:k-1})dx_{k-1}$$

The posterior PDF is calculated after the measurement becomes available with:

$$p(x_k | y_{1:k}) = \frac{p(y_k | x_k) p(x_k | y_{1:k-1})}{p(y_k | y_{1:k-1})}, \text{ where}$$

$$p(y_k | y_{1:k-1}) = \int p(y_k | x_k) p(x_k | y_{1:k-1}) dx_k$$

This is the optimal solution and has general character since the posterior density function cannot be calculated analytically in all cases.

The Kalman filter solves the equations analytically when the models are linear and the posterior PDF has Normal (Gaussian) distribution. The linear system obeys the superposition principle [2]. The model developed in 10.2 is non-linear, so the Kalman filter cannot be used. The extended Kalman Filter deals with non-linear models and it will be implemented here.

The extended Kalman filter (EKF) approximates the system to linear and then uses the analytical Kalman filter. This linearization is carried out with truncated Taylor series representation for the measurement model. The model is approximated to linear at the estimation point and therefore, the method is sub-optimal. More information can be found in [3].

The predicted value of Event Interval (EI) at k^{th} filter cycle is \hat{x}_k^- , called prior estimation. The minus sign at superscript expresses that the measurement at step k is not available, thus the prediction is done at state $k-1$. The estimation of EI after the measurement becomes available is \hat{x}_k^+ , thus the posterior estimation of EI. The plus superscript denotes that the measurement at k^{th} cycle is available.

The term prior estimation is used frequently than prediction in Kalman filter literature [3]. Both are synonyms in context of this thesis, but the term prior estimation is used here to be closer to the text sources. The posterior estimation is synonym to the estimation in the same manner [3, 2, 1].

Let us assume first that there is a relation between the measurement, prior estimation and posterior estimation. The relation is expressed generally by:

$$\hat{x}_k^+ = K' \cdot \hat{x}_k^- + K \cdot y_k, \quad \text{Equation 10.4}$$

where K and K' are time varying matrices and unknown at this point. The model in 10.2 has a single input variable, so x is number – matrix with 1x1 dimensions. Consequent, K and K' are single-sized matrices or simply coefficients.

Let \tilde{x}_k be the estimation error at step k . The prior and posterior estimation error can be expressed with:

$$\begin{aligned} \tilde{x}_k^+ &= x_k - \hat{x}_k^+ \\ \tilde{x}_k^- &= x_k - \hat{x}_k^- \end{aligned}$$

Substitution with Equation 10.3 and Equation 10.2 gives the result:

$$\begin{aligned} \tilde{x}_k^+ &= -x_k + K' \cdot \hat{x}_k^- + K \cdot f(x_k) \\ \tilde{x}_k^+ &= K' \cdot \hat{x}_k^- + K \cdot f(x_k) - \hat{x}_k^- - \tilde{x}_k^- \end{aligned} \quad \text{Equation 10.5}$$

The prime target is to find a posterior estimation with zero error. This means the real value can be calculated without an error after the measurement is available:

$$\varepsilon(\tilde{x}_k^+) = 0,$$

where $\varepsilon()$ denotes the estimation.

The Equation 10.5 with the zero estimation becomes:

$$(K' - 1)\mathcal{E}\{\hat{x}_k^-\} + K\mathcal{E}\{f(x_k)\} = 0$$

Let us denote the expectation of $f()$ as $\mathcal{E}(f(x)) = \hat{f}(x)$.

It is known $\mathcal{E}(\hat{x}_k) = \hat{x}_k$, then:

$$K' = \frac{\hat{x}_k^- - K \cdot \hat{f}(x_k)}{\hat{x}_k^-}$$

It gives the relation between K and K' . Substituted in Equation 10.4, one obtains:

$$\hat{x}_k^+ = \hat{x}_k^- \frac{\hat{x}_k^- - K \cdot \hat{f}(x_k)}{\hat{x}_k^-} + K \cdot f(x_k)$$

$$\hat{x}_k^+ = \hat{x}_k^+ + K \cdot (f(x_k) - \hat{f}(x_k)) \quad \text{Equation 10.6}$$

We shall use f instead of writing $f(x_k)$ for simplification further in the text.

The posterior error should be minimised in the second step regarding K . This is done with the minimisation of the standard deviation of the posterior error, thus the covariance matrix.

The covariance matrix P of the estimated prior error is:

$$P_k^+ = \mathcal{E}(\tilde{x}_k^+ \cdot \tilde{x}_k^{+T})$$

$$P_k^+ = P_k^- + K\mathcal{E}\left((f - \hat{f})(f - \hat{f})^T\right)K^T + \mathcal{E}\left(\tilde{x}_k^-(f - \hat{f})^T\right)K^T + K\mathcal{E}\left((f - \hat{f})\tilde{x}_k^{-T}\right) \quad \text{Equation 10.7}$$

where

$$P_k^- = \mathcal{E}(\tilde{x}_k^- \cdot \tilde{x}_k^{-T})$$

To minimize the error covariance (P_k^+) regarding K , the Jacobian must be minimized.

The Jacobian is

$$J = \text{trace}(P_k^+)$$

The derivate of J is set to zero. The solution gives the extreme point, which is the minimum value in this case. The following mathematical properties are used:

$$\frac{\partial}{\partial A} \text{trace}[A \cdot B \cdot A^T] = A(B + B^T)$$

$$\frac{\partial}{\partial A} \text{trace}[A \cdot B] = B^T$$

$$\text{trace}[C \cdot B^T \cdot A^T] = \text{trace}[A \cdot B \cdot C]$$

where A, B are matrices. The derivation of the Jacobian is:

$$J = \text{trace}(P_k^+) = \text{trace}(P_k^-) + \text{trace}\left[K\mathcal{E}\left((f - \hat{f})(f - \hat{f})^T\right)K^T\right] + \text{trace}\left[\mathcal{E}\left(\tilde{x}_k^-(f - \hat{f})^T\right)K^T\right] + \text{trace}\left[K\mathcal{E}\left((f - \hat{f})\tilde{x}_k^{-T}\right)\right]$$

The minimum of J regarding of K is at zero point of partial derivate.

$$\begin{aligned}
0 &= \frac{\partial J}{\partial K} = \frac{\partial}{\partial K} \text{trace}(P_k^+) = \frac{\partial}{\partial K} \text{trace}(P_k^-) + \frac{\partial}{\partial K} \text{trace} \left[K \mathcal{E} \left((f - \hat{f})(f - \hat{f})^T \right) K^T \right] + \\
&\quad + \frac{\partial}{\partial K} \text{trace} \left[\mathcal{E} \left(\tilde{x}_k^- (f - \hat{f})^T \right) K^T \right] + \frac{\partial}{\partial K} \text{trace} \left[K \mathcal{E} \left((f - \hat{f}) \tilde{x}_k^{-T} \right) \right] \\
&= \frac{\partial J}{\partial K} = \frac{\partial}{\partial K} \text{trace}(P_k^+) = 2.K \mathcal{E} \left((f - \hat{f})(f - \hat{f})^T \right) + \\
&\quad + \mathcal{E} \left(\tilde{x}_k^- (f - \hat{f})^T \right)^T + \mathcal{E} \left((f - \hat{f}) \tilde{x}_k^{-T} \right)^T \\
&= 2.K \mathcal{E} \left((f - \hat{f})(f - \hat{f})^T \right) + 2 \mathcal{E} \left(\tilde{x}_k^- (f - \hat{f})^T \right), \text{ where}
\end{aligned}$$

$$K = - \frac{\mathcal{E} \left(\tilde{x}_k^- (f - \hat{f})^T \right)}{\mathcal{E} \left((f - \hat{f})(f - \hat{f})^T \right)} \quad \text{Equation 10.8}$$

Substituting K in Equation 10.7 gives the following result

$$\begin{aligned}
P_k^+ &= P_k^- + \frac{\mathcal{E} \left(\tilde{x}_k^- (f - \hat{f})^T \right)}{\mathcal{E} \left((f - \hat{f})(f - \hat{f})^T \right)} \mathcal{E} \left((f - \hat{f})(f - \hat{f})^T \right) \left\{ \frac{\mathcal{E} \left(\tilde{x}_k^- (f - \hat{f})^T \right)}{\mathcal{E} \left((f - \hat{f})(f - \hat{f})^T \right)} \right\}^T \\
&\quad + \mathcal{E} \left(\tilde{x}_k^- (f - \hat{f})^T \right) \left\{ \frac{\mathcal{E} \left(\tilde{x}_k^- (f - \hat{f})^T \right)}{\mathcal{E} \left((f - \hat{f})(f - \hat{f})^T \right)} \right\}^T + \frac{\mathcal{E} \left(\tilde{x}_k^- (f - \hat{f})^T \right)}{\mathcal{E} \left((f - \hat{f})(f - \hat{f})^T \right)} \mathcal{E} \left((f - \hat{f}) \tilde{x}_k^{-T} \right) \\
P_k^+ &= P_k^- - K \mathcal{E} \left((f - \hat{f}) \tilde{x}_k^{-T} \right) \quad \text{Equation 10.9}
\end{aligned}$$

In order to calculate the expectation of $f()$ it is necessary to know its density functions. If the $p()$ is density function of $f()$ then:

$$\hat{f}(x) = \int_{-\infty}^{\infty} f(x)p(x)dx \neq f(\hat{x})$$

The PDF is not analytically defined, thus the equation cannot be solved directly. It is approximated instead. A Taylor series is used to calculate an approximation. The Taylor series for $f()$ is defined as:

$$z(x) = \sum_{n=0}^N \frac{1}{n!} \cdot \frac{\partial^n f(x_o)}{\partial x^n} \cdot (x - x_o)^n$$

With a higher degree N , the function $z()$ approaches $f()$ at point x_o . For $x_o = \hat{x}^-$ the series is developed as:

$$f(x) = f(\hat{x}^-) + \frac{\partial f(\hat{x}^-)}{\partial x} (x - \hat{x}^-) + \frac{1}{2} \frac{\partial^2 f(\hat{x}^-)}{\partial x^2} (x - \hat{x}^-)^2 + \dots \quad \text{Equation 10.10}$$

Taking an expectation of both sides gives:

$$\begin{aligned}\varepsilon(f(x)) &= \hat{f}(x) = f(\hat{x}^-) + \frac{\partial f(\hat{x}^-)}{\partial x} \varepsilon(x - \hat{x}^-) + \frac{1}{2} \frac{\partial^2 f(\hat{x}^-)}{\partial x^2} \varepsilon((x - \hat{x}^-)^2) + \dots \\ &= f(\hat{x}^-) + \frac{\partial f(\hat{x}^-)}{\partial x} \varepsilon(\tilde{x}^-) + \frac{1}{2} \frac{\partial^2 f(\hat{x}^-)}{\partial x^2} \varepsilon(\tilde{x}^{-2}) + \dots \\ \varepsilon(f(x)) &= f(\hat{x}^-)\end{aligned}$$

The result is substituted in Equation 10.8:

$$\begin{aligned}P_k^+ &= P_k^- - K \varepsilon \left(\left(\left[f(\hat{x}^-) + \frac{\partial f(\hat{x}^-)}{\partial x} \tilde{x}^- + \frac{1}{2} \frac{\partial^2 f(\hat{x}^-)}{\partial x^2} (\tilde{x}^-)^2 + \dots \right] - f(\hat{x}^-) \right) \tilde{x}_k^{-T} \right) \\ &= P_k^- - K \varepsilon \left(\left(\frac{\partial f(\hat{x}^-)}{\partial x} \tilde{x}^- + \frac{1}{2} \frac{\partial^2 f(\hat{x}^-)}{\partial x^2} (\tilde{x}^-)^2 + \dots \right) \tilde{x}_k^{-T} \right)\end{aligned}$$

The first two terms of the Taylor series can be truncated and then we obtain:

$$P_k^+ = P_k^- - K \frac{\partial f(\hat{x}^-)}{\partial x} P_k^- = (I - K f'(\hat{x}^-)) P_k^- \quad \text{Equation 10.11}$$

The PDF is approximated to a linear function through the truncated Taylor series. The integral can be analytically calculated. Substitution of Equation 10.11 in Equation 10.8 gives the result for K , which can be analytically calculated:

$$K = - \frac{\varepsilon(\tilde{x}_k^- (f - \hat{f})^T)}{\varepsilon((f - \hat{f})(f - \hat{f})^T)} = \frac{P_k^- \frac{\partial f(\hat{x}^-)}{\partial x}}{\frac{\partial f(\hat{x}^-)}{\partial x} P_k^- \frac{\partial f(\hat{x}^-)}{\partial x} + v} = \frac{P_k^- f'(\hat{x}^-)}{f'(\hat{x}^-) P_k^- f'(\hat{x}^-)^T + v} \quad \text{Equation 10.12}$$

The posterior maximal error is used instead of the posterior standard deviation. This must be done since the standard deviation of the measurement model is unknown, see 10.2. The posterior maximal error, thus the UI with EI, is approximately proportional to standard deviation. Consequent, the resulting standard deviation of the Kalman filter is approximately proportional to the standard deviation of the predicted value. There is no impact on the M-LU implementation because the M-LU is interested in an estimation value. The semi standard deviation gives the degree of believe in the prediction.

Using the truncated Taylor series in Equation 10.6 provides an approximation for update estimation:

$$\hat{x}_k^+ = \hat{x}_k^+ + K (f(x_k, \hat{x}_k^-) - \hat{f}(x_k, \hat{x}_k^-)) = \hat{x}_k^+ + K (f(x_k) - f'(x_k)) \quad \text{Equation 10.13}$$

The state estimate propagation is

$$\hat{x}_k^- = A \hat{x}_{k-1}$$

The prior error covariance can be expressed by the prior error:

$$\begin{aligned}\tilde{x}_k^- &= x_k - \hat{x}_k^- = A x_{k-1} - A \hat{x}_{k-1} = A \tilde{x}_{k-1}^- \\ P_k^- &= \varepsilon(\tilde{x}_k^- \tilde{x}_k^{-T}) = A \varepsilon(\tilde{x}_{k-1}^- \tilde{x}_{k-1}^{-T}) A^T = A P_{k-1}^- A^T + \sigma\end{aligned} \quad \text{Equation 10.14}$$

The derivation of the extended Kalman filter for M-KE is concluded with this final step.

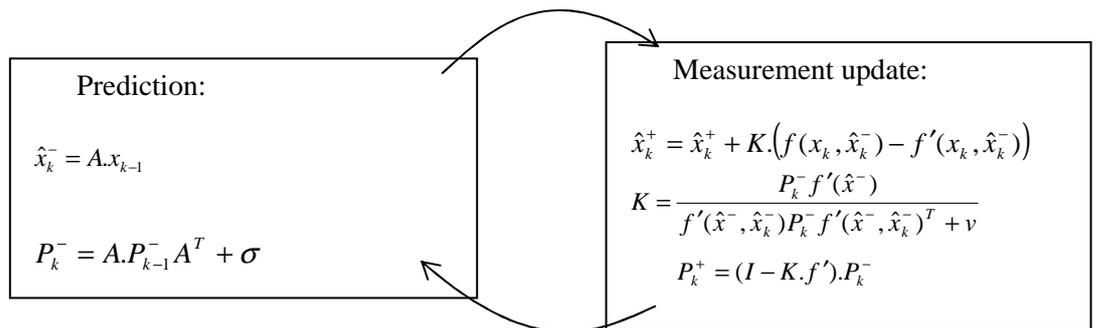
The result can be summarised by:

Model:

$$x_k = Ax_{k-1} + w \quad w = N(0, \sigma)$$

,

$$y_k = f(x_k, \hat{x}_k^-), \text{ with error } v(x_k, \hat{x}_k^-)$$



10.8 Simulation

The simulation is a proof of concept showing the qualities of the new algorithm. The simulation is carried out with Matlab 7.0 on PC 1GHz, RAM 512 MB. The objectives were to gain accuracy of the prediction and estimation. The processing time is out of scope.

The differences between the real and the simulation deployment are shown in Figure 10.4 and Figure 10.5. In real deployment, the prior estimated EI (predicted) is calculated using the extended Kalman filter see Figure 10.4. The UTP are calculated using the transformation function. The update procedure is executed until ETP happens. Then the filter cycle begins again. In contrast, the simulation is performed in the reverse way see Figure 10.5. The absolute EI are generated. For every absolute EI is generated the UI with ETP. The gathered statistic is analysed at the end. It must be underlined that a remainder must be used to generate EI from the absolute EI. The remainder is described in 8.5

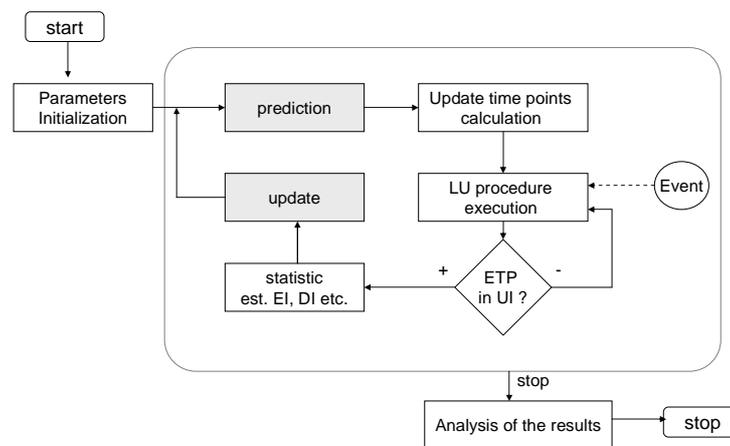


Figure 10.4: General structure real time deployment

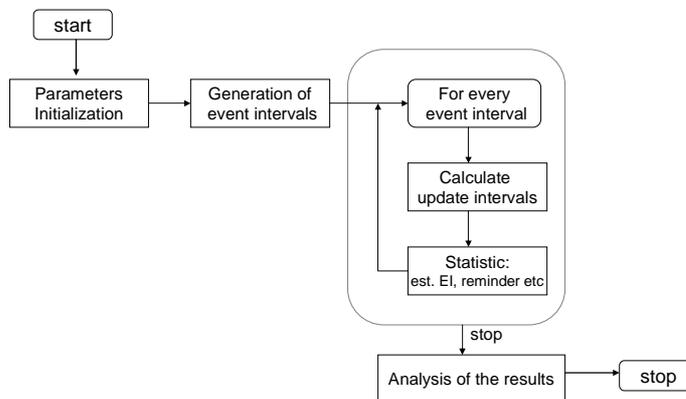


Figure 10.5: Offline simulation structure

The statistical variables are the same as for the Particle filter in 8.5 and Fuzzy controller in 9.4. The variables are described in 7.5.2 and are: {mean DI}, {mean UI}, {max UI}, {max user def DI}, {const UI}, {const UI mean DI}, {const UI / mean UI} and {LU < const UI}. The statistics are presented in three plots generated in the same way as in the previous methods.

10.9 Simulation results

The same equations and parameter, see 7.5.1, are used for the generation of the absolute EI as in Particle filter (see 8.6) and Fuzzy controller (see 9.5). The results can be compared in this way. The first four simulations are carried out on 6000 Event Intervals with MDI of 5 sec. The real data is used in the last (fifth) simulation with 2500 Event Intervals and 300 sec of MDI.

10.9.1 White noise

The generated values for the absolute Event Intervals and the estimated ones are shown in Figure 10.6. The estimated and real Event Intervals almost overlap which speaks for the good qualities.

The results histogram is presented in Figure 10.7. The Update Intervals with ETP are concentrated in both extremes. There is a high concentration at the MDI, which shows poor prediction. On the other hand, there is a concentration at the small intervals, which corresponds to a good prediction. The mean UI by constant update is 4.11 sec and it is 1.53 times bigger than the new method using the Extended Kalman filter (see 8.6.1). The new algorithm shows an outperformance of about 153%, considering the means. More than 62.4% of all updates were smaller then the constant Update Intervals.

It can be concluded that the method has a good performance on shifted white noise. The performance is worse than the Fuzzy controller (section 9.5.1) and the Particle filter (section 8.6.1). The Update Interval with Event Time Point is shown in Figure 10.8. The errors are systematic, thus there are not region with high errors. The good results can be explained because the input data (signal with white noise) correspond the chosen variable model, i.e. the random walk model.

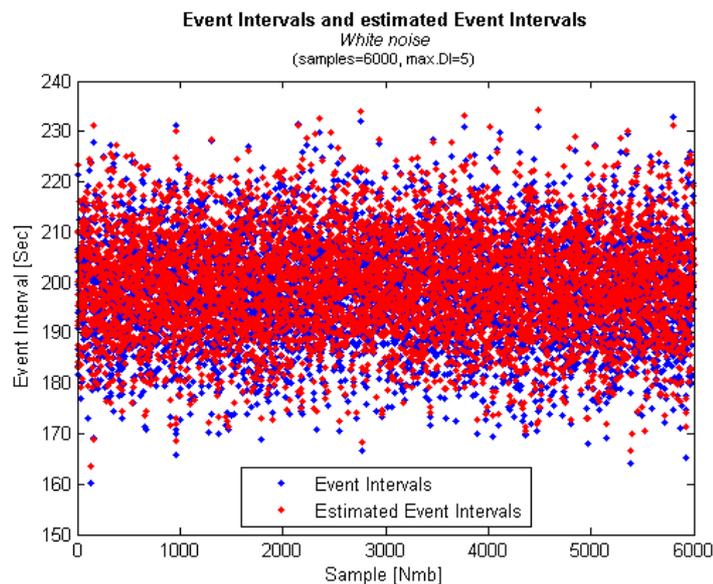


Figure 10.6: White noise, EI and estimated EI

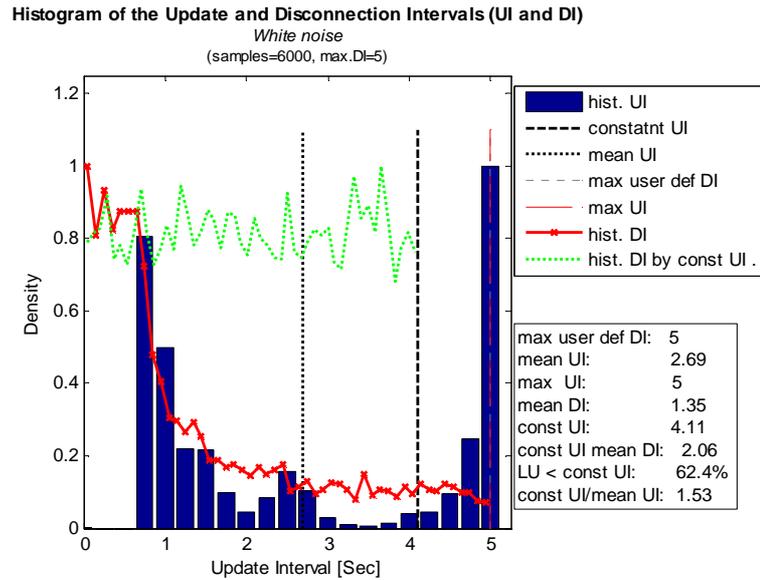


Figure 10.7: White noise, results histogram

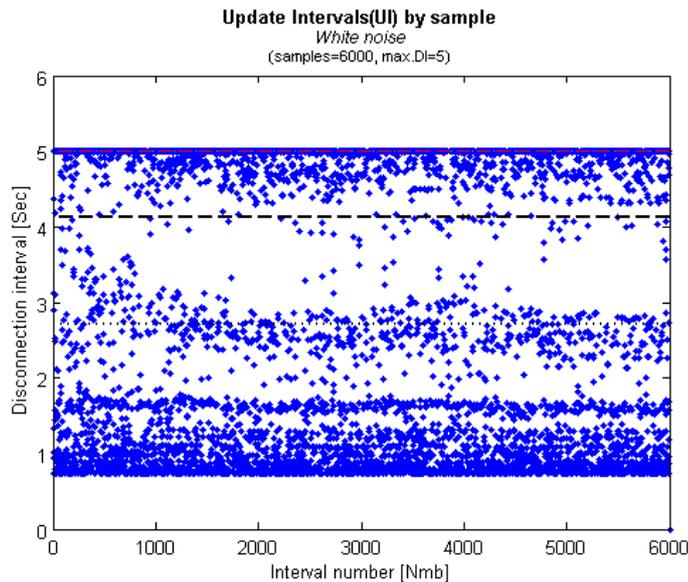


Figure 10.8: White noise, Update Intervals with ETP

10.9.2 Two rotating white noise sources

The absolute Event Intervals are generated with two rotating shifted white noises sources. The data is already described in 8.6.2. The estimated and real Event Intervals are shown in Figure 10.9.

This case is very problematic for the new method. It can be concluded from the results histogram in Figure 10.10 that the new method does not have any advantages when compared to the constant update. About 50% of the updates are better than the constant update, which is not advantage at all (50% are worst). The constant Update Intervals with the same resources are smaller than the new method (0.79). The constant update is better than the new method when one considers the mean Update Interval. It is obvious that the new method has a problem in this case. The reason is the form of the PDF, which is not similar to the bell curve,

see Figure 10.11. The random walk model approximates to bell curve, which is definitely not optimal for this simulation data.

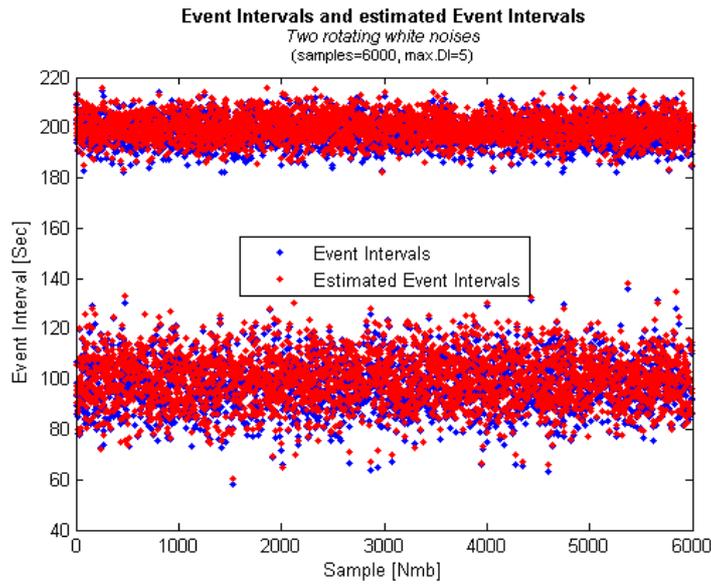


Figure 10.9: Two rotating noise sources, EI and estimated EI

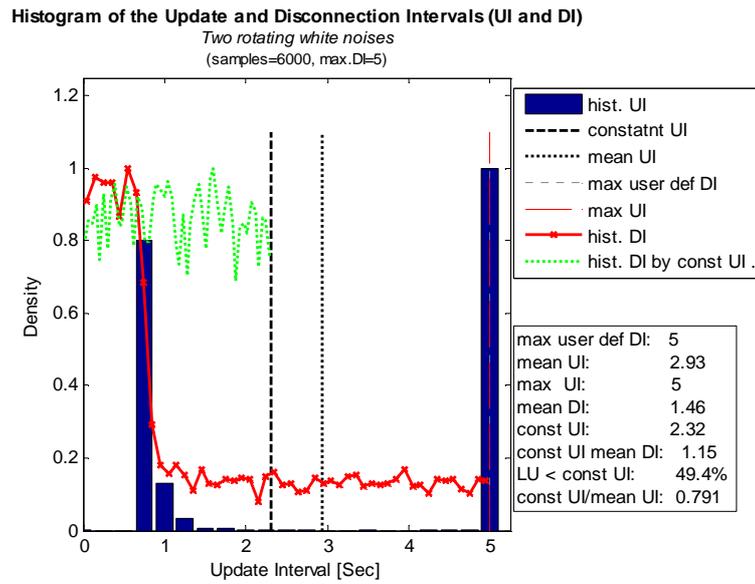


Figure 10.10: Two rotating noise sources, result histogram

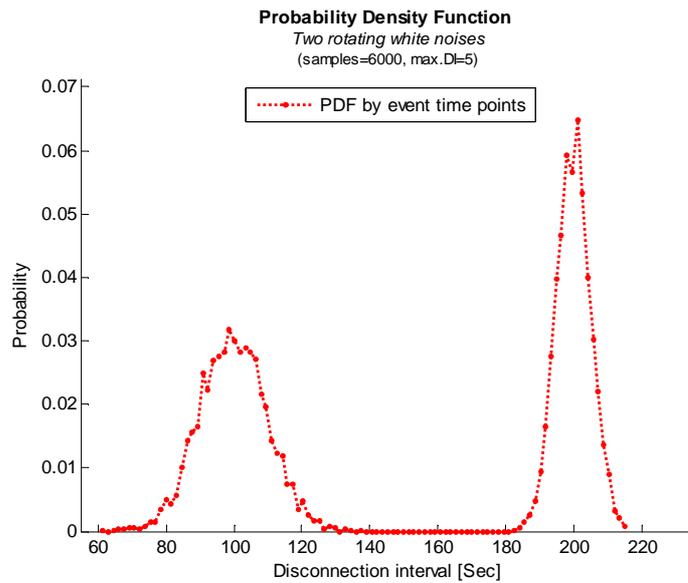


Figure 10.11: Two rotating source noises, PDF of the Event Intervals

10.9.3 Sinus-based EI with white noise

The absolute Event Intervals are calculated using a sinus-based function with white noise. The equation is specified in 8.6.3. The generated Event Intervals and estimated intervals are plotted in Figure 10.12.

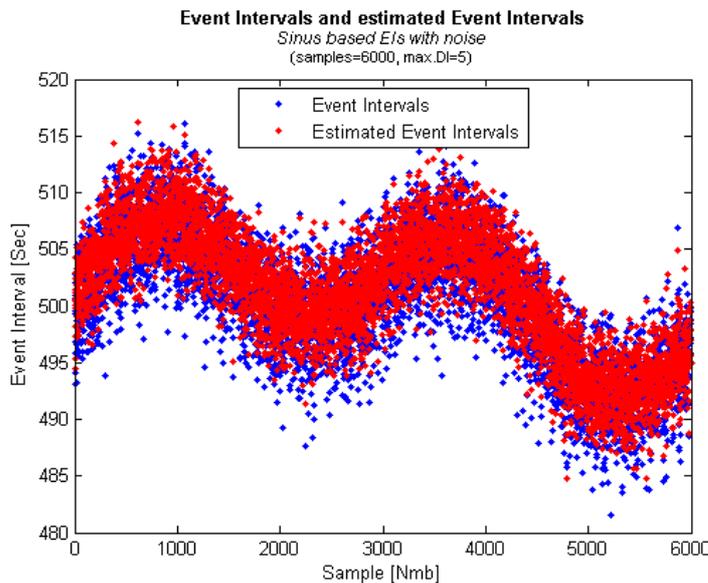


Figure 10.12: Sinus based with noise, EIs and estimated EIs

The results in Figure 10.13 show excellent qualities of the method. In over 92.5% of the updates, the method is better than the constant updates with the same resources. The mean UI with the new method is smaller than the constant updates by a factor of 2.63. There is clear outperformance of the new method. The result histogram has higher concentration at the left side (small values). This is a sign for good prediction. The method is better than the Fuzzy controller in 9.5.3 and worse than the Particle filter with the same data, see 8.6.3. It must be underlined the Fuzzy controller had not received successful training and this probably explains the poor results.

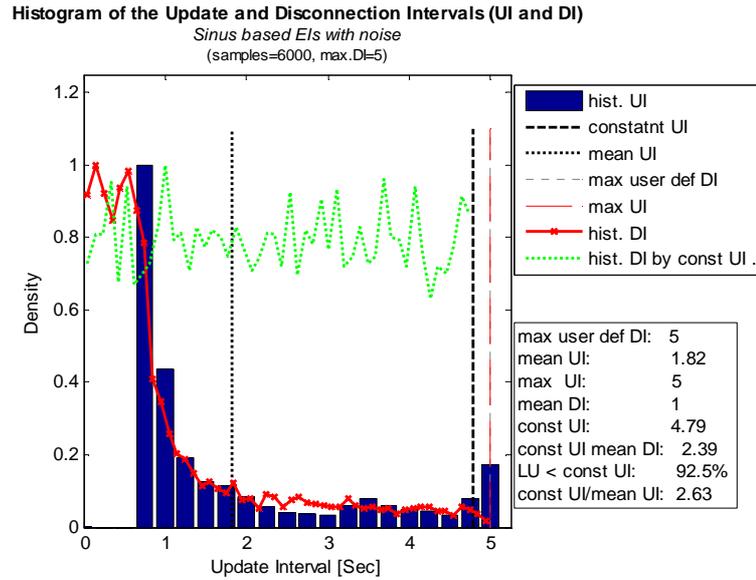


Figure 10.13: Sinus based with noise, result histogram

The Update Intervals with ETP are shown in Figure 10.14. It is interesting to notice that the intervals are dependent on the derivate of the Event Intervals, thus the change of the curve direction causes a change in the prediction error.

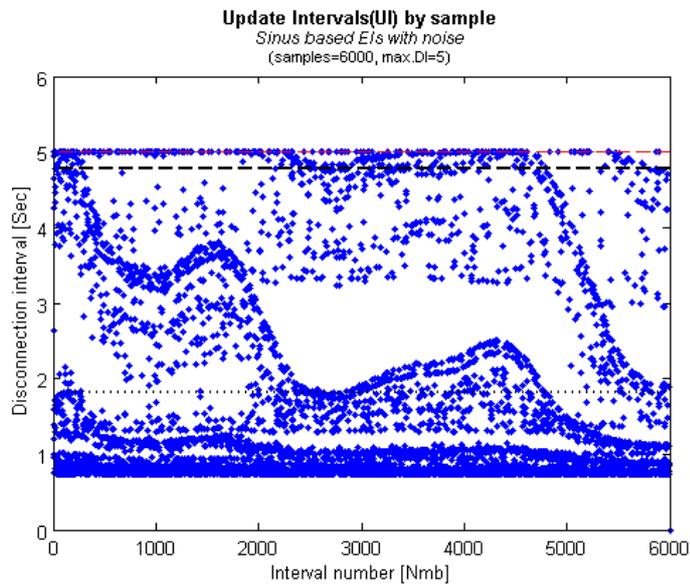


Figure 10.14: Sinus based with noise, Update Intervals with ETP

10.9.4 Non-linear EI with white noise

Non-linear absolute EI are generated by recursive equation with added noise at each interaction as described in 8.6.4. The Event Intervals are shown in Figure 10.15. The results in Figure 10.16 show a histogram with a nice distribution with maximums at the extremes. The histogram is higher at small intervals, which means that the algorithm has produced a good prediction of EI. 69.5% of the updates are smaller than the constant updates. The mean UI is 1.45 times smaller than the constant one. The extended Kalman method delivers good results, but they are poorer than the Fuzzy controller and Particle filter.

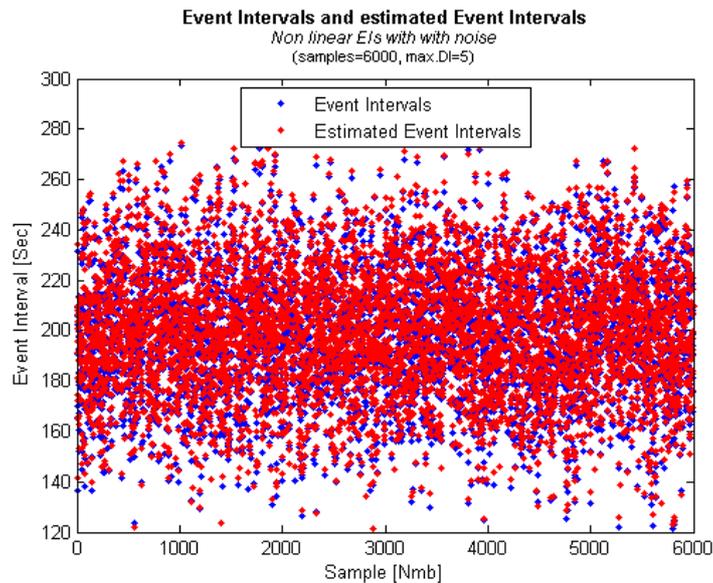


Figure 10.15: Non Linear recursive data, EIs and est. EIs

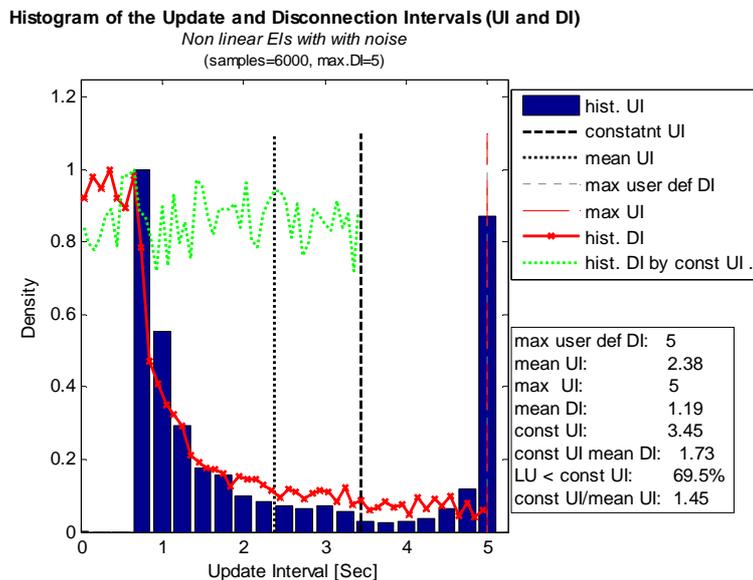


Figure 10.16: Non linear recursive data, result histogram

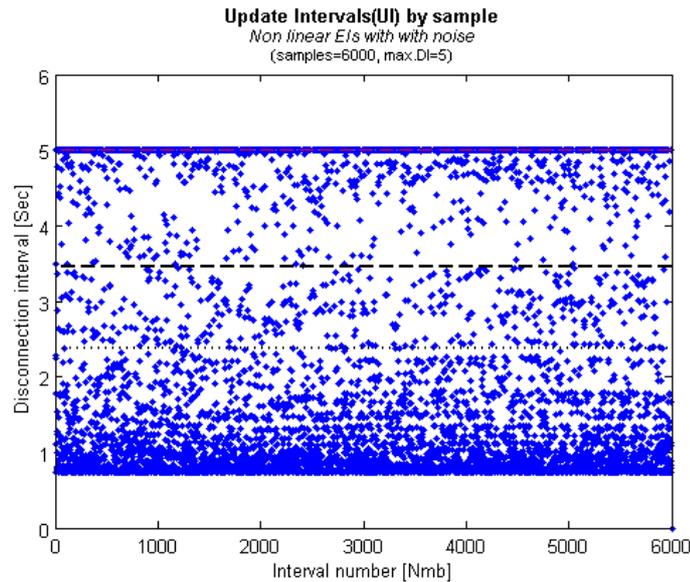


Figure 10.17: Non linear recursive data, update intervals with ETP

10.9.5 Real data

The last simulation is real data with 2500 Event Intervals and maximum Disconnection Interval of 300 sec. The data represents the successful login of dial up uses over 5 days. The same data is used as in the previous chapters for Fuzzy controller and Particle filter. The Event Intervals have day cycle properties and the EI are set relative to midnight. The pre-processing is described in 8.6.5.

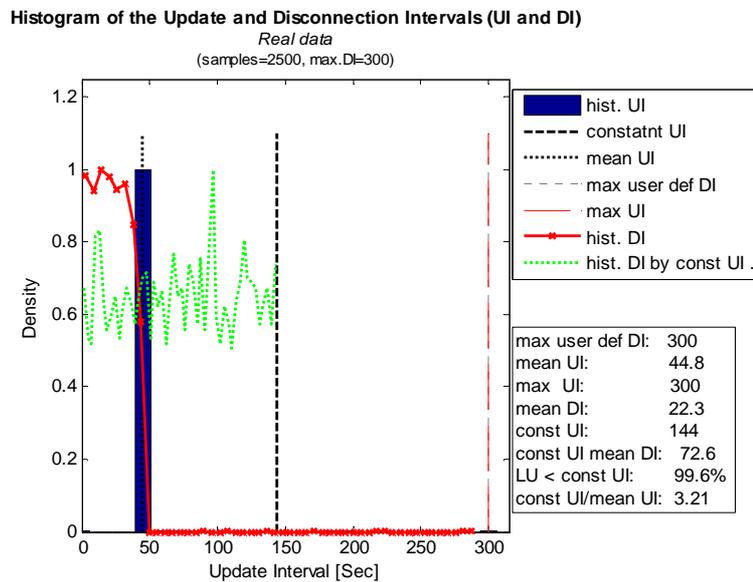


Figure 10.18: Real data, result histogram

The results are excellent, see Figure 10.18. The extended Kalman filter achieves smaller Update Intervals with ETP in 99.6% of the samples. The mean Update Intervals are 3.21 times smaller than the constant update. The suggested method is better than the Particle filter and Fuzzy controller. The algorithm solves the real data task excellently, although this is considered as the most difficult case. The real case can be compared to a smooth changing signal (such as sinus) with some white noise. This is treated better by the Kalman method

than by any other method. The UIs with ETP are presented in Figure 10.19. The Figure 10.20 shows the PDF.

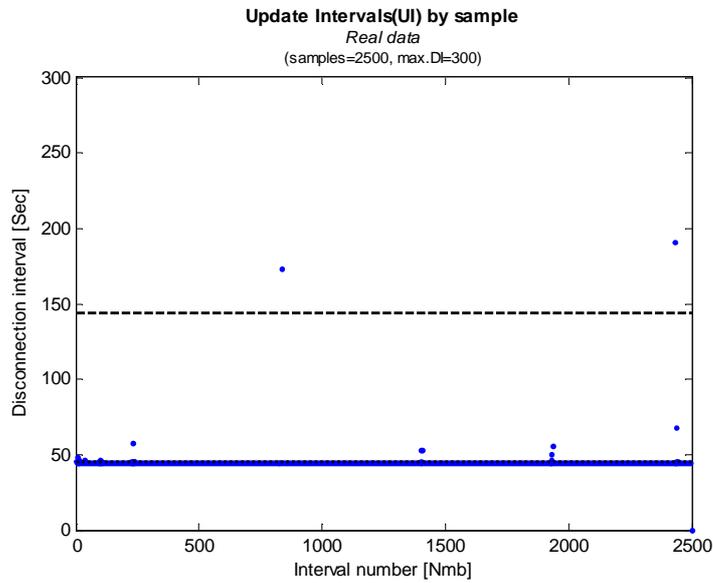


Figure 10.19: Real data, UI with ETP by sample

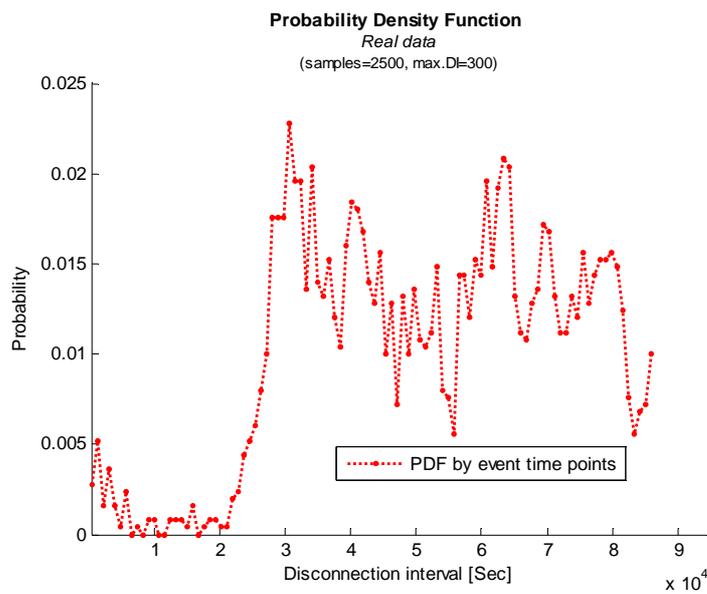


Figure 10.20: Real data, PDF

10.10 Conclusion and future work

The suggested method has an impressive performance in the most important simulation run with real data. It is even better than the Fuzzy controller (chapter 9) and Particle filter (chapter 8). On the other hand, the algorithm shows poor performance in the case with two rotating white noise sources (section 10.9.2). There is no constant performance in all cases. The real data covers a case with a typical employer day cycle. Every algorithm tested (see chapters 8 to 10) has its advantages and disadvantages. There is no universal algorithm and a winner cannot be chosen. The exact EI behaviour can be used for choosing the algorithm.

An important advantage of the EKF algorithm is that it requires only one parameter before starting and namely the Maximum Disconnection Intervals. The Fuzzy controller and Particle filter require a number of parameters, such as the number of rules, particles etc. These parameters must be set to optimal values. Otherwise, the performance becomes poor. There is no need of evaluation of optimal parameters in the suggested method.

There are still open points for future work, which can improve the performance:

- Using overlay cycle behaviour, such as week cycle and day cycle for example. Currently, the algorithm can only perform one type, such as day. Implementation of a multiply cycle will improve the performance, like considering weekday and day hour in the same time.
- The cross point of the linearisation equals predicted EI, thus $l = P_k^+$, see 10.3. The Update Intervals begin to decrease by the beginning of the updates. Setting the cross point dependent on the semi standard deviation can improve the performance. The non-linear part should be close to the estimated Event Interval for small prediction errors and vice versa. This is more an optimisation task of the algorithm.
- Other transformation functions can be developed and tested

10.11 References in chapter 10

- [1] Arulampalam, Sanjeev, Maskell, Simon, Gordon, Neil, Clapp, Tim, "A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking (2001), IEEE Transactions on Signal Processing
- [2] Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems", Copyright © 1960 by ASME
- [3] Gelb, Arthur, "Applied Optimal Estimation", MIT Press, 1974, ISBN: 0262570483
- [4] Wiener, N., "The Extrapolation, Interpolation and Smoothing of Stationary Time Series," John Wiley & Sons, Inc., New York, N.Y., 1949.
- [5] Maybeck, Peter S, Academic Pr "Stochastic Models, Estimation and Control.", Mathematics in Science and Engineering, Juni 1979
- [6] Grewal, Mohinder S. und Andrews, Angus P., "Kalman Filtering. Theory and Practice using MATLAB: Theory and Practice Using MATLAB", Wiley & Sons, Januar 2001
- [7] Tzvetkov, Vesselin, "SIP registration optimization in mobile environments using extended Kalman filter", IEEE ChinaCom, August 2008

11 Application of M-LU in external protocols

The M-LU can contribute to all protocols that have an update mechanism, which can be abstracted to the M-LU location update. The update efficiency can be improved without any significant change in the protocol's specification. The M-LU defines the intervals for executing the update procedure.

The requirements on the protocol are that the update procedure must be executable in variable intervals and there must be a Boolean feedback of the execution. Through the update procedure, the status is proactively verified. The initiator asks for the status and the responder replies as shown in Figure 11.1. Important is that the status (feedback) is Boolean, thus true or false. Depending on the feedback, some actions are taken at the initiator. It is irrelevant what type of status information is requested. For example: The update can be the verification whether or not the connection is still active. If there is a reply, it is active and if not, then some actions must be initiated.

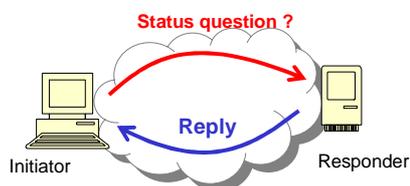


Figure 11.1: Abstraction of the update procedure

Every procedure, which can be abstracted in this way, is a candidate for the deployment of the new framework. It can be implemented in routing, security, application protocols etc. Some examples of possible applications are provided in the following sections.

11.1 Optimisation of Dead-Peer-Detection in IKE using M-LU

The Internet Key Exchanged protocol [2], described in A.2.2, includes a mechanism for detecting dead peers (DPD). It is specified in [1] and consists of two notification messages: "R-U-THERE" and "R-U-THERE-ACK". The initiator sends the "R-U-THERE" message to verify, whether the connection is still active. The receiver replies to it with "R-U-THERE-ACK". The time interval of DPD is constant, for example every 120 sec. The DPD is executed in idle time only, where no traffic is exchanged.

The Event Time Point in M-LU corresponds to failure of the DPD execution, thus missing "R-U-THERE-ACK" reply. In the location update procedure, *true* corresponds to receiving DPD reply and *false* corresponds to missing DPD reply. The DPD is very similar to the M-LU update procedure. Using the history of DPD failures, it is possible to achieve much more efficient update intervals. The algorithm suggested for M-LU can be implemented in a straightforward way. There is no need to change the DPD specification. The method is described by the author in [3].

11.2 SIP optimisation through M-LU

SIP [4] is the most widely implemented protocol for VoIP communication. Almost all Service Providers offer telephone services based on SIP. The SIP has two methods for dealing with dead peers, NAPT router and firewalls: registration and options request.

The first method requires frequent registrations of the SIP client. The registrar sets the validity period for every registration. Before the registration expires, the client must register

again. The SIP registrar updates the client's PoA parameter by every registration. If a client's PoA has changed then it will be updated by the next registration. After a PoA change, the client is unreachable until it reregisters. The registration generates packet exchange, which updates the NAPT binding at the same time. When the SIP registrar uses this method, the typical registration intervals are 90 sec. To apply M-LU protocol, the SIP registrar must track the changes of client's PoA by comparing the current and last values. It adjusts the registration intervals by setting the expire attributes. The registration interval corresponds to the Update Intervals in M-LU terminology. The Event Time Point corresponds to a change of the client's PoA. The UI with ETP is the registration period in which the PoA has changed.

In opposite to the M-LU, it is not the client but the server that sets the registration period, i.e. the Update Interval. The deployment is very simple, since there is no change in the standard. It is sufficient to extend the server to use M-LU algorithm.

The second possibility for updating the PoA is the SIP Options [4] request. The SIP registrar or the SIP client sends proactively empty options request according [4]. The receiver replies to the message with its status. The initiator gets feedback whether the connection is still active or not. The Event Time Points are the failures to receive a reply, i.e. the client or registrar has changed its IP/UDP parameter. The request intervals can be optimised depending on the reply status.

The M-LU implementation in SIP will reduce the number of messages processed by the SIP server (proxy or registrar). This will directly improve the efficiency and reduce the investment see [8] and 11.2.1.

11.2.1 Resources consumed in SIP

To obtain a feeling for the wasted resources let us consider a real example: VoIP Service Provider (SP) with 200 000 online clients per SIP server host (proxy or registrar). The registrar sends an empty option header (dead-peer-detection mechanism in SIP) every 20 seconds to check the status of every client. Let us consider the minimal load threshold (the best case), thus these dead-peer-detections are equally distributed in time. The server must process 10 000 updates per second in order to keep the status of all clients up to date. The usual performance of the server machine with a simple firewall and connection processing can handle maximum of about 200 000 packets per second (Test made on Dual Intel Xeon 2.4GHz see [7]). This means that about 5% of the server utilization will be consumed by updates. The example shows, that the network and CPU resources for updates must not be underestimated.

11.3 Binding update message in Mobile IP

Mobile IP [6, 5] can benefit from the M-LU protocol for optimising the Binding update procedure described in 2.10.1.1. The Mobile Node registers its current location to the Home Agent or/and to the Correspondent Node. The binding update procedure will be performed typically at certain intervals. To integrate M-LU, the binding update procedure must be extended to deliver a feedback to the sender if the PoA has changed. The Home Agent or Correspondent Node must reply to the Mobile Node if the IP/UDP parameters have changed. This information only requires one bit, which must be carried in the reply. A new payload type can also be implemented.

The execution of the binding update corresponds to the Update Interval in M-LU. The change of Mobile Node's PoA maps to the Event Time Point. The Mobile Node can implement M-LU with the optimisation of its Update Intervals in this way.

11.4 References in chapter 11

- [1] Huang, et al., "A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers", RFC 3706, February 2004
- [2] Kaufman, C., "The Internet Key Exchange (IKEv2)", RFC 4306, December 2005
- [3] Tzvetkov, Vesselin, "Optimization of Update Intervals in Dead-Peer-Detection using adaptive Fuzzy Logic", AINA-2007, IEEE Conference
- [4] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002
- [5] Johnson, D. et al., "Mobility Support in IPv6", RFC 3775, June 2004
- [6] Perkins, Ed., "IP Mobility Support for IPv4", RFC 3344, August 2002
- [7] Kadlecsek, József, Pásztor, György, "Netfilter Performance Testing", www.netfilter.org, 2004
- [8] Tzvetkov, Vesselin, "SIP registration optimization in mobile environments using extended Kalman filter", IEEE publishing pending, 2008

12 Conclusion

Secure IP mobility is not a trivial task and has many aspects, as shown in chapter 2. The potential solutions have many deficits presented and analysed in 2.10. They do not cover the requirements on secure IP mobility.

The first group of solutions consists of mobile and secure protocols at different layers (see 2.10.1). Since the security protocol runs over the mobility layer, the signalling is not sufficiently protected. The signalling is protected only by the mobility build-in methods, as for example the authentication in MIPv4. This is the major disadvantage. The second group brings mobility features to the secure protocols (see 2.10.2). This solution is getting much closer to the desired target. The main issues are caused by the core protocols design. The original protocol (IKEv2) is static and is not designed to work in a mobile environment. Despite adding some extensions for mobility, the protocol is always far from optimum in terms of packets formats and state behaviour (see 2.10.2). The third group contains research projects on mobility covering certain security aspects (see 2.11). They have slightly different targets, for example assuming IPv6 or homogeneous infrastructure with certain features at every host. These are interesting concepts targeting Intranet solutions of IPv6 networks. The problem with tracing user movements is unsolved in all approaches.

12.1 Development of Mobile VPNs

A new protocol, called M-VPN is developed and specified in this thesis, since the existing approaches cannot fulfil the requirements defined in chapter 2. The new protocol has a protection against movements tracing, protection of mobile signalling and of application data. A further major objective is to develop a practical protocol, which is not over-engineered and easy to integrate into the existing infrastructure. The development of Mappers gives the advantage of physical separation between the Mobile Node, Tunneling Node and application. The M-VPN integrates excellent in the existing Authentication, Authorization and Accounting (AAA) structures. The M-VPN uses existing cryptographic techniques such as AES, RSA, DH, which assures cryptographic robustness of the algorithms. Because of the similar structure to IPSec and TLS, is not expected to have any vulnerability (see chapter 6).

12.2 Mobile Location Update protocol

The major part of this thesis is a mathematical model for the location update (M-LU). The protocol is part of the M-VPN and answers the question when to send a location update. The question is very important in mobile environments. Constant updates, like every 10 seconds, are not efficient for mobile environments. They lead to unnecessarily resource consumption and poor performance.

In this thesis, a novel framework is developed with a mathematical solution for the update problem. A model representing the update procedure in M-VPN was developed, which enables the use of known mathematical methods. Based on classical theory, three types of solutions were developed:

- Sequential Monte Carlo Sampling (see chapter 8) is a solution constructing the probability based on history of PoA changes.
- Adaptive Fuzzy controller (see chapter 9) is based on Fuzzy logic for dealing with multivalent values, suitable for subjective description.

- Extended Kalman Filter solution (see chapter 10) uses the popular extended Kalman filter, which is a fully analytical derived sub-optimal solution.

The contribution of the M-LU is not limited to M-VPN alone. The novel framework can be implemented in a variety of protocols for improving their performance, see chapter 11.

12.3 Simulation results

An important part of every new development is its simulation. It provides a proof of concept showing the qualities of the new method. Simulation of M-LU protocol is conducted and presented in 8.5, 9.4 and 10.8. The simulation was carried out in Matlab 7.0 with a standard PC. The results are compared to constant updates involving the same resources. Furthermore, the comparison between the suggested methods was also made, i.e. Fuzzy controller, particle Filter and Extended Kalman filter.

Many simulations were executed to test the qualities of the new methods. Five of them were chosen as representative for the majority of scenarios. The first four use pseudo random input values generated with Matlab. The last one uses real data gathered from the network statistics for dial up access.

The new methods outperform the constant update. There are fewer resources consumed for the same disconnection interval. In practical terms, the new methods require less CPU and network resources and achieve less disconnection. It is difficult to choose a winner between the three methods. Different algorithm can be favoured depending on the usage. A general recapitulated can be made as follows:

- Fuzzy controller has excellent performance when some expert knowledge has already been gathered. The expert knowledge can be easily integrated in profiles, like typical employee day cycle (working hours, days etc.).
- The Monte Carlo Sampling is suitable when multiple sources for the PoA changes are involved. The PoA changes can be linear and not linear, like changes caused by user movement and NAPT devices at the same time. The method can rebuild sophisticated Probability Density Functions (PDF).
- The extended Kalman filter gives excellent results for typical employee usage without any prior knowledge. It performs very well on Mobile Nodes with abrupt changing behaviour, like workers with early and late shifts. The method has poor performance for complex PDF with multiple sources of PoA change.

12.4 Future work

The simulation clearly shows the advantage of the new algorithms. In all cases, there was better performance than the constant UI. There are still areas in which the algorithms can be improved and they are part of future work:

- Optimisation of the input parameter must be implemented in real deployment. These are selected systematically in the simulation through multiple tests. The optimization can be made using automatic control engineering methods. The input parameters for optimisation are the number of: particles, added updates, Fuzzy rules etc.
- The cycle events, such as day, week etc, can be considered in overlay. Currently the algorithms apply day cycle. The consideration of week and year may improve the performance.

- In Fuzzy controller, training should be implemented during the operation. The suggested method in chapter 9 can be extended, so that the rule base is updated after every update/prediction cycle. The algorithm can handle a dynamically changing PDF function in this way. Furthermore, the risk of missing rules decreases, because of a steady feedback of the rule base (see 9.5.3). The implementation must regenerate the Fuzzy rules after every value received.
- The PDF can change over the time. A forgetting factor of PDF can be added to the Particle Filter in chapter 8.
- Transformation functions in extended Kalman Filter in chapter 10 can consider other cross point in the approximation. They may be depended on semi standard deviation, see 10.3.

A Appendix - Internet structure and relevant protocols

In order to examine the mobile and secure topic, it is very important to understand the Internet from technical perspective. The Internet structure has changed a bit over the last decade. The new protocols must consider the non-transparent access via NAPT. This chapter provides a description of the IPsec and NAPT, since they are very relevant for secure IP mobility. The here provided description is not part of the common literature and cannot be found in RFC standards in this form. This chapter focuses on implementation properties, deployments, popular not standardized features and drafts. Without understanding these properties of IPsec and NAPT, it is not possible to design robust secure IP mobility. The text is organised as follows: the Internet access is presented in A.1, the NAPT is described in A.2.1 and the IPsec is the scope of A.2.2.

A.1 Internet access

The Internet is not only growing but also evolving - the structures and protocols are steadily changing. The design targets and objectivities set decades ago cannot be met in the current Internet. In order to bring clarification to all readers, a brief overview is provided in this section.

The Internet can be structured in different ways depending on the abstraction, like on dedicated OSI layer or depending on the technology. The ISPs (Internet Service Provider) classify on the purpose: access network, core network with its aggregation and backbone part.

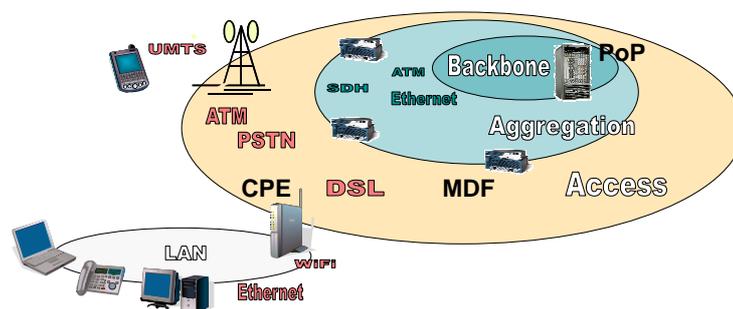


Figure 12.1: Access, Aggregation and Backbone network

The Local Area Network (LAN) is the network connecting the end devices, like PCs. Typically, this can be WiFi [42] or Ethernet connection (Figure 12.1). The LAN is controlled by the end user. The Gateway for the LAN hosts is the Customer Premises Equipment (CPE). The access network stretches from the CPE to the Main Distribution Frame (MDF) where the ISP's active hardware is installed. The connection is mostly copper wire (twisted pair) used for aDSL connection. The outer part of the core network is the aggregation network, which starts by MDF and finishes at Point of Presence (PoP). It concentrates the customer's physical connection on one single cable. The customers are separated logically in VLAN [45], VCP/VCI on ATM [47] etc. The backbone network is between the PoPs. It transports IP packets, without any customer separation. The different network areas are shown in Figure 12.1. The different ISPs backbones are connected at Interconnection Exchange Points, so the packets are routed globally. Currently, there is trend for moving the active hardware nearer to the customer in Serving Area Interface to achieve higher speeds.

During the last decade, Internet was accessed mainly by fixed cable connection or through the Public Switched Telephone Network (PSTN) with modems (narrow band access). The end hosts were assigned public IP addresses. The hosts are connected transparent to

Internet, because bidirectional communication was possible without any restrictions. There were not NAPT (see definition in A.2.1) device. The layers of the Internet access through PSTN and by fixed cable connections are shown in Figure 12.2. The AL

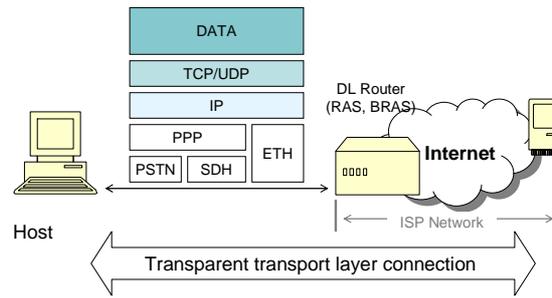


Figure 12.2: Transparent Internet connection without NAPT

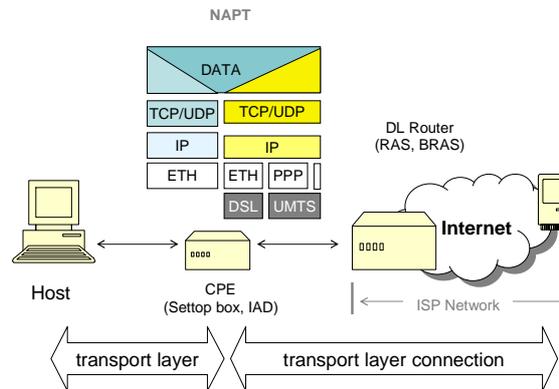


Figure 12.3: Non transparent Internet access through NAPT

The Internet is accessed nowadays in non-transparent way. The end host obtains private IP address, which is not routable and reachable from the Internet. The Network Address and Port Translation (NAPT) [1] is implemented in order to allow the communication with Internet. The general layer structure is shown in Figure 12.3. The major Internet access scenarios regarding the IP connection are:

UMTS (3G) and GPRS (2,5G)

The Universal Mobile Telecommunications Systems (UMTS) or shortly 3G enables fast IP Internet access for cell phones [40, 39]. The majority of modern cell phones support this standard and there is network coverage in the urban areas. The UMTS is a lower layer protocol and does not define the IP topology. The IP network is designed by the Internet Service Provider (ISP).

The mobile devices are assigned private IP addresses in most of the cases. The ISPs involve NAPT gateways, which are part of the ISP network and cannot be influenced by the end customer. The consequences of NAPT for the hosts are described in A.2.1. The use of private IPs is not part of the UMTS standard but a common policy of the ISPs. Some ISPs use public IPs for business services. Other providers restrict the usage of the Internet to certain applications, such as HTTP. In this way, they reduce the bandwidth demand and consequently offer cheap flat rates.

Wireless LAN (WiFi) and Hotspots

The Wireless LAN [42] or shortly WiFi is very popular technology for uncomplicated connection of devices. The majority of the modern CPEs and laptops support this technology. Typically, the ISP assigns one public IP address to the CPE device. It is shared between multiple end devices connected with WiFi (Wireless LAN). In order to enable Internet access, the CPEs involve NAPT [2]. The end devices use private IP addresses, which are translated to single public IP address. If the customer has some control over the NAPT router, he can configure static NAPT bindings (see A.2.1).

DSL Router and Cable Modems (TV Provider)

The DSL routers use private addresses on the LAN side and one WAN (Wide Area Network) public address. The router can also contain Wireless LAN access. The Wireless LAN is simply another router interface and the principle is the same as Hotspots in terms of IP transparency. The Internet access is also through NAPT.

Set-Top Boxes and Integrated Access Devices (IADs)

Set-top boxes integrate functions such as video over IP, video on demand etc. The IADs have mostly VoIP functionality. They enable advanced services and are currently a target of ISP's. These services require extended network features such as quality of service and traffic shaping etc. There are usually dedicated logical connections between the set-top box (CPE) and the ISP access network for every service. The logical separation enables flexible network topology. When the customer accesses the Internet behind the set-top box, NAPT is in between.

Corporate and Campus Access

Corporations and campuses operate on LAN/MAN, where the participants use major Intranet services. The users access the Internet using proxies or NAPT: the proxies provide application layer Internet access. The application proxies are more restrictive and therefore very popular in corporations. The Internet access can be also provided via a statefull inspection firewall and NAPT. Users of these networks suffer from the same disadvantages as exist when accessing the Internet behind an NAPT device.

A.2 Overview of relevant protocols

NAPT is major part of the Internet access and therefore, must present in details. The IPSec is used to secure most of the existing mobile solution, thus playing a major role.

A.2.1 NAPT Overview

Network Address and Port Translation [1, 2, 3, 4] is wide spread technology for connecting different IP realms, which are not directly routable. Through the translation of the IP/UDP header, both realms are able to establish indirect IP connection. The common implementation is the translation of private IP addresses used in LAN to public IP routable on the Internet. Translations of public IP to public IP or private IP to private IP are also possible.

There is no consensus about the exact terminology, so clarification of the notation shall be provided here. The Network Address and Port Translation (NAPT) [1, 2, 3, 4] are synonyms for other popular notations such as: NAT, PAT, Masquerading, Port forwarding,

static/dynamic NAT. The NAPT is a more general term and covers all facets of translation, thus with or without port translation and static or dynamic. It is used in this document.

NAPT was first mentioned in 1994 by Kjeld Egevang and Paul Francis in [1]. The original motivation was to develop a technique that directly addressed the accelerated consumption of IPv4 addresses. The NAPT was to bring temporary relief during the development of IPv6: “provide temporarily relief while other, more complex and far-reaching solutions are worked out” [7]

More than 12 years later, NAPT is implemented in all Internet capable devices and operating systems: Microsoft XP/2K, Linux, DSL Router, WiFi Hotspots etc. The pressure for the successor protocol IPv6 is low and the target has been met.

A.2.1.1 NAPT Operation

The NAPT device intercepts the IP packets and replaces the header and payload values, so that the IP packet can be forwarded to/from certain host. Let us concentrate on the translation private IP to public IP in the common DSL Router access scenario. In Figure 12.4, the NAPT device intercepts the packets from A to B and translates the private IP 10.0.0.1 in 82.0.0.1. The new IP (82.0.0.1) address must be routable to the NAPT device from the Internet.

NAPT table entry, called *mapping* or *binding*, is created for every session. The mapping (binding) is protocols dependent and can contain the IP and port parameters. The information in the entry (binding) must be sufficient for de-multiplexing the received packets to the right session. The combination of all parameters in the mapping entry must point to a unique session. Otherwise, the packet coming from the Internet cannot be mapped to the private IP.

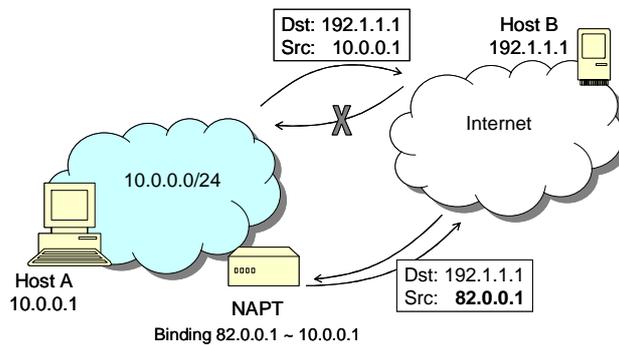


Figure 12.4: Connecting IP realms with NAPT

Without a NAPT, the packets can be routed to a destination (Figure 12.4 presents Host B with IP 192.1.1.1), but cannot be returned to the sender. The private IP’s overlap (are used multiple times), so there is no unique proprietor host.

The NAPT device has interfaces in both IP realms (private and public in our example). The interfaces are known as *inside* and *outside* interfaces. The translation can be carried out prior to forwarding or after forwarding. This is defined by configuring the execution of translation at the inside or outside interface. For example, the DSL routers execute NAPT on the outside WAN interface. When packets from the Internet are received, the translation is executed prior to forwarding. The packets arriving on the LAN side are forwarded and then translated.

The NAPT device translates the header values and the payload. A different procedure has been developed for every protocol. A comprehensive overview of the IP/TCP is provided at the following Figure 12.5. The translation procedure exchanges the address and port values.

Consequently, the checksum field must be recomputed. The replaced fields are marked (in blue and red) in the Figure 12.5.

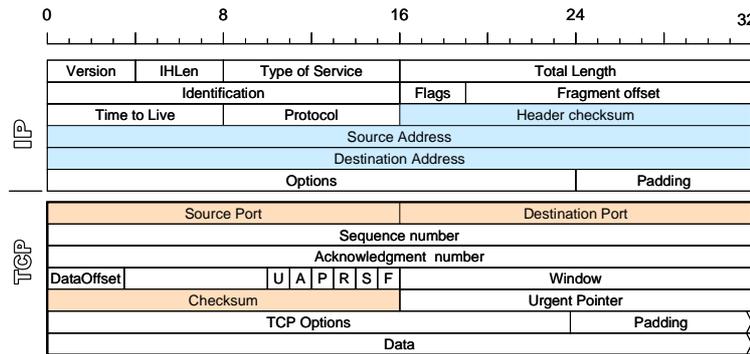


Figure 12.5: IP and TCP header overview in NAT

A.2.1.2 NAT Types

There are different types of NAT, depending on the translation:

- 1) *static NAT*. The NAT mapping is constant and pre-configured at the device. This type is also referred in literature as port forwarding, when referred to TCP or UDP (Figure 12.6) The translation can involve only IP addresses independent of the TCP/UDP ports, as shown in Figure 12.4. The main property of the static translation is that no session state monitoring is needed. The connection is bi-directional and every packet received can be translated without the knowledge of the UDP/TCP connection state. This feature is widely used for gaming and file sharing etc. The main condition is that the translation mapping must be exactly predefined, which is not always possible.

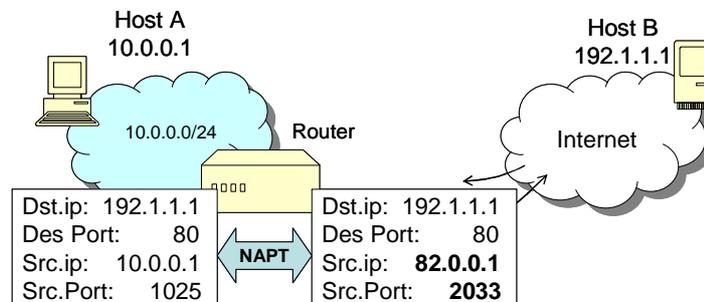


Figure 12.6: Translation of IP address and port

- 2) *dynamic NAT*. The entry is dynamic when some of NAT mapping parameters cannot be predefined. The missing parameters are extracted from the packet headers during the establishment. In literature, this is known also as masquerading or PAT. The best example is the NAT usage in the home network at DSL router. The host used random source ports by establishing a session. The port is unknown in advance. Therefore, the exact mapping cannot be predefined. During UDP/TCP establishment, the port information is extracted from the packet (on-the-fly) to create the binding. After closing the session, the entry is removed and the resources become available for new translation.

A.2.1.2.1 Dynamic NAPT issues

There is a problem when to remove the NAPT mapping once created. The unused mappings must be removed. Otherwise, the device and public resources will be overloaded. (The sense of NAPT is to aggregate resources through sharing). The main idea is the remove the mapping when the communication session is closed, thus the mapping is not used. The problem is when the session is closed. The UDP is connectionless protocols and there is not defined session finish at transport layer. In this situation, it must be used session definition at application layer. This is not always possible, for example: HTTP is sessionless protocol. A TCP session cannot be properly closed because of hardware failures.

To handle this, an idle timeout is added to every NAPT entry. Idle timeout is generally used for all NAPT mappings. Typical values are 5 minutes for UDP and 15 min for a TCP connection. Unfortunately, a timeout increases the possibility of erasing connections still in use. For example: the hosts are not exchanging packets, but they need the connection. This is a major problem for protocols. In order to keep the entry, the applications use keep-a-live mechanisms described in section A.2.1.8.

The connection using dynamic NAPT is unidirectional. The connection can be established from one side only. For example: Only on the LAN side in the popular DSL implementation. It is not possible to establish connection from the Internet. The advantages and disadvantages for the applications are discussed in chapter A.2.1.12.

A.2.1.3 NAPT implementations

There are different NAPT implementations having major influence on the functionality of protocols. Some implementations are friendly to definite protocols and some are not. Considering the Figure 12.3, the binding at CPE is created when PC sends packet to some Internet host. The Internet host responds to PC behind the NAPT and the NAPT device must de-multiplex the packet. The implementations for de-multiplexing are:

- 1) *Symmetric*: The NAPT entry matches the local host address, source port number, destination address and port number. Any change to one of these fields requires a different NAPT binding. This is the most restrictive implementation.
- 2) *Full-cone*: This is the least restrictive NAPT implementation. It establishes the mapping matches, when the destination IP and source port are correct. The source port and IP address are not considered.
- 3) *Restricted-cone*: The de-multiplexing with the NAPT entry is done by comparing the destination port, destination IP address, source IP address, but not the source port.
- 4) *Port-restricted-cone*: The matching criteria are the destination IP and port and the source port. The source IP address is not considered for the translation.

A.2.1.4 Simple Traversal of UDP through NATs (STUN)

The applications have different compatibility to the different NAPT implementations. An application must determine: first, if it is behind an NAPT device and secondly, what type of NAPT policy is implemented. For this reason, a Simple Traversal of UDP through NATs (STUN) [16] was developed. The protocol is widely used in VoIP clients.

In general, the STUN defines the client-server (initiator-responder) relation. The client wishes to locate the NAPT device and the server assists this. The principle is that the client sends a request and the server answers. The server must be transparently connected to Internet. Currently, there are many STUN servers in the network.

The client sends an initial request packet to the STUN server. The server responds with the IP address and port of the received packet in the payload. The client compares the payload values with the original ones. If the payload values and the original values are the same then

there is no NAPT in between. There is an NAPT device in between if the returned values differ.

Involving the similar logic of request and responses, the client can determine most of the NAPT types. The client cannot determine whether there are multiple NAPT devices in between. Multiple NAPT routers are simplified to one. Unfortunately, STUN cannot detect all types of NAPT. For example, whether it is a static or dynamic NAPT. The worst case is always assumed for the application, thus dynamic NAPT.

A.2.1.5 NAPT payload translation and bundled sessions

Many protocols, such as SIP [50], FTP [49], NetBios etc. consist of bundled TCP/UDP connections. The packet payloads of the original session contain IP addresses and ports of the following sessions. The applications use IP/port values in the payload to establish new sessions. It's very important for the NAPT device to translate the IP and port values in the payload. Leaving this information unchanged leads to a failure of the protocol. The receiver tries to establish a connection to an unreachable IP and port from the payload in original session.

A practical case is demonstrated on the FTP [49] protocol in Figure 12.7, but the principle is the same for all bundled sessions. The FTP client establishes a signalling session to the FTP server in active mode (the working principle is different in passive mode). We assume there is an NAPT router in between, as in Figure 12.7. If the client executes “ls”, “get” or “put” commands, a new data session must be established from the server to the client, in which the data is transferred. The IP address and TCP port for the data session are defined by the client. The values are sent in the payload of the signalling session. The client asks the server to establish a TCP session to its specific local IP and port (active FTP). The client is behind an NAPT device, so it will use its private addresses and some open local port. Using these values, the server cannot establish the session, since the private address is not reachable from the Internet. The data transfer fails if the payload is not translated. In practical terms, the clients can login but cannot transfer any data, so there is no real use for the connection.

In order to handle session bundles, the NAPT devices must also translate the payloads with the correct addresses. This requires that the NAPT device is aware of all protocols and knows which fields are to be translated. The encoding of the IP address and port in the payload is carried out in different way. For example: some of them can contain the parameters as string values or as binary bytes sequences. The task of the NAPT device becomes very difficult.

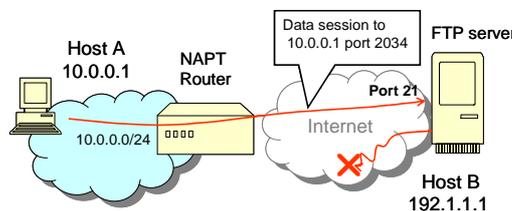


Figure 12.7: FTP data transfer

The correct translation of the payload is not sufficient for working with bundled sessions. NAPT mappings (bindings) serving following sessions must be created in advance using the information in the original session. In the FTP example (Figure 12.7), the NAPT mapping must be set for the data session. The data session is established from the Internet to the host. Without the mapping, the packets will be dropped. The translation strategy is different for every application protocol.

The translation is not required by some protocols, such as the STUN protocol. If the NAPT device translates the STUN payloads, then the client cannot always determine that there is NAPT in between

Modern protocol designers are aware of this short-coming in NAPT, so they try not to create bundled sessions or transmit IP or port information in the payload. This principle is also used in this thesis, as framework requirement for protocol design.

A.2.1.6 Working with NAPT

NAPT has strong associations with a client-server model of communications. As long as the servers have constant IP addresses and the clients initiate the TCP connection, NAPT will work without any significant problems. For client-server applications without bundled sessions, NAPT is semi-transparent without any limitation on the protocol. The widespread deployment of NAPT and the continued use of client-server applications attest its capability to perform transparently and effectively.

Peer-to-peer applications and bundled sessions are highly problematic for NAPT, because they do not match the NAPT design model. The peer-to-peer (P2P) protocols require transparent bidirectional connection between the hosts. The connections can be established by both peers, there is no strict requirement for the side of initiation. In order to solve these issues, all P2P applications use the following methods:

- 1) The P2P networks involve nodes with NAPT proxies functionality. The NAPT proxy remains between the peers. The communication is split in two client-to-server connections. The peers always communicate indirectly through proxy. Each peer contacts the NAPT proxy, which must be transparently reachable from the Internet. The NAPT proxy transports the data received from one connection to the other as shown in Figure 12.8. The NAPT proxy notation is different between the applications, for example in SIP it is RTP proxy or media gateway. Skype(TM) uses the term super node.
- 2) The second possibility is changing the original protocol to become dynamically client-server and/or to use NAPT implementation properties. Using protocols like STUN, the peer is able to find out which type of NAPT is involved. The applications can open some port, to be reachable from Internet if there is a full-cone. Additionally, the application can dynamically check if one of the participants is transparently connected. This host can be used as the server. Modern P2P sharing protocols like DirectConnect try to use this type of option. They are client-to-server protocols but, abstracted to higher layer, they can be considered as P2P network.

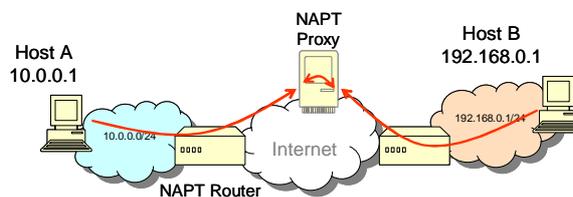


Figure 12.8: NAPT proxy

A.2.1.7 NAPT with ICMP, ESP and other transport layer protocols

The NAPT mapping usually involves IP and port values. Protocols like ICMP[51], ESP[21] do not contain port information in their header, since they are network layer protocols. These types of protocols are very challenging. NAPT requires a unique header values for de-multiplexing the return packets. The destination IP address is equal for all

packets received from the outside interface. This is insufficient to match the right session. The NATP device uses different mapping policies for every transport protocols to solve this issue. For ICMP, the NATP device usually adds the original ICMP header in the packet payload. The returned packet includes the original header and it can be correctly translated back by the NATP device. The SPI number is used with the ESP protocol. This is unique part of the ESP header. The problem is that this type of binding creation must be defined for every protocol and it does not cover all cases see A.2.2.10.6.

A.2.1.8 NATP Keep-Alive

When the hosts are behind dynamic NATP, the bindings are removed after a certain idle timeout (see A.2.1.1). When an active binding is removed, the host becomes unreachable. The NATP device cannot translate the packets and the packets are dropped. The applications use Nat-Keep-Alives (updates) in order to avoid this type of problems. The updates are usually empty packets including headers only. The only purpose is to keep the NATP binding active. The updates packets are handled at the NATP routers, which reset the idle timer.

The Nat-Keep-Alive mechanism must not be mixed with Dead-Peer-Detection (DPD A.2.2.10.2). The Dead-Peer-Detection (DPD) detects an unreachable end host. It does not target any NATP binding update. Although, its side effect is the update of the NATP entry.

A.2.1.9 NATP and fragmented IP packets and PMTU

NATP has significant problem with the fragmented packets. When the host sends large packets on a link with a smaller MTU size [52] the packets are typically fragmented. This is carried out by either the sender or the intermediate router. A property of fragmented packets is that the transport header is contained only in the first fragment. The following fragments are chained by their fragment ID only. There is not TCP/UDP header in the IP fragments. When the NATP device receives packets (fragments) without transport headers, the binding cannot be matched. There are two strategies available to overcome this problem:

- 1) The NATP device reassembles the fragment as the end host does. They perform the NATP translation after the original IP packet has been reassembled with all its fragments. The packets are typically fragmented again after the translation. Assembling all fragments is question of resources at the NATP device. There is a need of a buffer to save the fragments until the whole packet is available. The assembling of all packets slows down the connection, as all fragments must be available before proceeding. The end-to-end delay increases.
- 2) The NATP device uses the fragment ID's and IPs to map the fragments to certain session. Forwarding is performed using the fragment ID's. The advantage of this behaviour is that it works significantly faster than the previous one and does not require additional resource for buffers. Unfortunately, it has major problems: the checksum of the header must not be used; out-of-order fragments are dropped.

A.2.1.10 NATP and illusion for security

Many users consider NATP as security feature and not network issue. Probably, because it is was introduced in this way by some marketing people. In fact, NATP has hardly any advantages for the end customers. It brings a lot of restrictions when working with P2P protocols. It is advantageous for the ISPs: there is no need for additional IP addresses for all end devices.

The perfect NATP device is invisible to the participants. It should not restrict the communication at all if possible. The perfect NATP provides no protection to the

communication. In the use of full-clone implementations, the Internet host can easily reach the inside hosts with full scanning.

It is very important to understand that NAT is not a security mechanism and if there are some security properties, they are side effects and depend strictly on the implementation. Firewalls are the solution for increasing the security network.

A.2.1.11 NAT and the security protocols

The use of NAT is problematic for the security protocols, which use the IP addresses in their payloads or protect the IP/UDP headers. It is a principle question: On the one hand, the security protocols try to uncover any manipulation of the IP/UDP header and block potential attacks. On the other hand, NAT manipulates the header and this is necessary for the connectivity. There are big issues with IPSec protocol, described in following chapter A.2.2.10.1.

A.2.1.12 The controversy of NAT

NAT is a very controversial function. It helps the administrator and ISPs to grow their networks without changing the routing tables and without providing more IP addresses. This keeps the resources and investments very low. Furthermore, the end customers are no longer restricted to having just one computer, since the ISP assigns just one IP. Most DSL customers have multiple devices, like laptops, PCs, PDAs etc.

NAT brings enormous restrictions to IP communication. The end host is unaware of its communication address. Many IETF standards are drifted away from the reality, since the assumed transparent bi-directional connectivity is not present. Some of the protocols become useless. There are many documents written about NAT properties [9, 10, 11, 12]. The major disadvantage of NAT can be summarised as:

- The translation at the NAT device is unknown to the participants. The hosts are unaware of their translated IP/port. The IP/port changes without any notification.
- Bundled sessions require payload translation, which is not always supported by the devices.
- Dynamical NAT does not support applications where the initiator is outside. Peer-to-peer services, like VoIP, cannot function in an NAT environment.
- The P2P applications are forced to deploy different architecture with external proxy. This leads some further centralisation of services and leaving the P2P principles.
- The behaviour of NAT varies dramatically from one implementation to another. It is very difficult for the applications to predict or expose the precise behaviour of NAT, which may exist in between.
- Robust security in IP environments typically operates on an end-to-end model, where both ends include additional information in the packet to detect manipulation of the packets. NAT changes the IP and TCP/UDP header values. If the security protocol protects against the manipulation the IP and TCP/UDP header, the NAT device translation will be treated as an attack. It is not possible to use protection of IP and TCP/UDP in an NAT environment.
- NAT have no inherent failover. NAT is an active in-band mechanism that cannot fail into a safe operating fallback mode. When a NAT goes offline,

all traffic through the device is dropped. An NAPT device is a single point of failure

- NAPT sit on the data path and attempt to process every packet. Obviously, there are issues regarding the bandwidth scaling.
- With NAPT there is no clear, coherent, and stable concept of network identity. From the outside, these NAPT-filtered interior devices are visible only as transient entities.
- Policy-based mechanisms based on network identity, like Policy Quality of Service [QoS]), cannot work through NAPT.
- Normal forms of IP mobility are broken when any element behind the NAPT attempts to roam beyond its local private domain. Solutions are possible, generally involving specific NAPT-related alterations to the behaviour of the Home Agent and the mobile device.
- NAPT may drop IP fragments in either direction: without complete TCP/UDP headers, the NAPT may not have sufficient stored state to undertake the correct header translation.
- NAPT do not support mobility, like Mobile IP

A.2.2 IP Security (IPSec) and IKE

IP Security (IPSec) [20] and IKE/ISAKMP [34, 35] are widely used protocols for delivering protection at the IP layer. The IPSec is used in majority of mobile solution, like Mobile IP, and therefore, it is important for this topic. The relevant properties are highlighted here. There are two version, thus IKEv2 [34] and IKEv1 [35]. The IKEv1 is the most used version having implementation in all OS. The IKEv2 can be considered as consolidation of additional RFCs in the IKEv1. There are less implementations of IKEv2 at this point. This section concentrates in IKEv1, since the author describes practical properties of existing implementation.

The IPSec implementations are also popular as Virtual Private Networks (VPN) or Layer 3 VPN. The expression VPN is far more general and it is not correct to use it as a synonym for IPSec. The VPN defines the possibility of overlaying networks on virtual basis. They can be realized with various technologies, like MPLS [43], VPLS [44, 46], VLANS [46], IPSec [20] etc.

A.2.2.1 IP Security Overview

IPSec (IP Security) is a protocol for protection at IP layer, which can include different transport protocols and different algorithms. It is standardised by IETF in [20]. IPSec is a peer-to-peer protocol, thus the communication participants have the same administrative right and duties. The features delivered by the protocols are:

- data integrity
- access control (authentication and authorisation)
- optional data authentication
- optional confidentially
- replay protection

IPSec is not a single protocol but a suite (collection) of security protocols for security. The architecture is defined in RFC 4301 [20] as follows:

- Security protocols- Authentication header (AH [19]) and Encapsulation Security Protocol (ESP [21])
- Key management – IKE [34, 35] and Oekey [27]
- Algorithms for encryption and authentication, like 3DES [24]

The relation and interaction between the IPSec protocols is complex and the different standards cannot be considered independently of each other. An understanding of the general protocols can be achieved only through understanding the interaction between the different elements. IPSec has the rum as complex protocol, although with approximately one dozen RFCs, it is quite compact. For example, SIP [50] includes more then fifty RFCs. The reason for this notorious image of IPSec is probably the compactness and short additional explanation in RFC texts.

The exchanges packets can be protected in two ways – Authentication Header (AH) and Encapsulation Security Payload (ESP). The differences are that AH protects the IP header and does not deliver confidentiality. The ESP is achieves confidentiality through encryption and does not protects the outer IP header. The ESP and AH require pre configured secure keys. They can be delivered through the ISAKMP protocol or can be configured manually. The ISAKMP is the common way for delivery the key material for ESP and AH.

Both the ESP and AH protocols can be used in transport and tunnel modes. In transport mode, the ESP/AH header is between network layer (IP) and transport layer (TCP/UDP). The data following after the ESP/AH header is protected. In tunnel mode, the original IP packet is encapsulated in the ESP/AH. A new outer IP header is generated. They are two IP headers outer and inner in tunnel mode. The outer is called gateway IP address. The gateway provides the cryptographic functions and (de) encapsulates the packets. The inner IP is the end host, which do not need any IPSec functionality in this way.

A.2.2.2 Encapsulation Security Payload (ESP)

The ESP protocol [21] uses block cipher, like AES in CBC [22] mode of operation, to encrypt the content. The integrity protection is optimal thought truncated HMAC with MD5 or SHA1. The key in HMAC achieves authentication of the packet. Sequence number is used for replay protection.

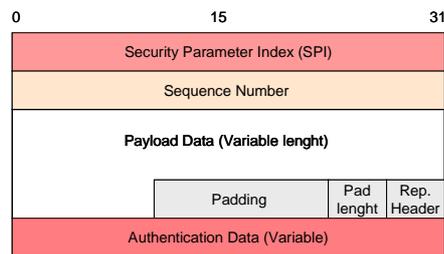


Figure 12.9: ESP Header

The ESP header is presented in the Figure 12.9. The Security Parameter Index (SPI) field is the unique connection indication. Using the SPI field, the packet is matched to the right session. The SPI number are part of the IPSec database, called SAD. In the SAD all parameter are contained which are required for the decryption and processing of the packet, as explained in A.2.2.4. The second field sequence number indicates the sequence of the sent packet. It is used for protection against replay attacks and described in chapter A.2.2.7. The payload data follows, for example the TCP header with the application data. The padding is added to achieve the requisite data length for use with block encryption algorithms. The last field is the authentication field with truncated HMAC (Authentication Data). The HMAC is 12 bytes. The size of ESP header varies due to the padding and encryption algorithm. It is usually between 80 and 120 bytes.

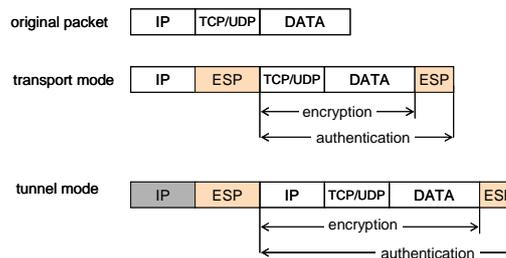


Figure 12.10: ESP protection

The ESP is used in tunnel and transport modes. The protected areas in both tunnel and transport mode are shown in Figure 12.10. In tunnel mode, the original IP header is protected and cannot be read by intermediate authorities, for example NAPT. The outer header is not protected, so it can be read and manipulated. This is a very practical advantage of ESP when used in NAPT environments.

A.2.2.3 Authentication Header (AH) protocol

The AH protocol [19] achieves data authentication, reply and integrity protection of the packet. Unlike the ESP, it does not provide confidentiality. The header information is very similar to the ESP header and it is presented in Figure 12.12. The Figure 12.11 presents the AH in transport and tunnel mode.

The AH is not delivering confidentiality and have problem with intermediate NAPT. Probably because of that, AH is not very popular. The AH and ESP protocols can be used together to increase integrity protection, although this is not implemented by the vendors in practice.

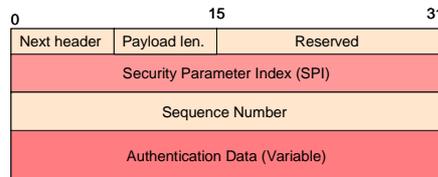


Figure 12.12: AH header

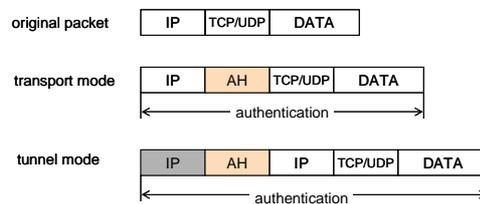


Figure 12.11: AH tunnel and transport mode

A.2.2.4 Security Policy Database (SPD) and Security Association Database (SAD)

The IPsec implementations have at least two databases according to the RFC [20] - Security Policy Database (SPD) and the Security Association Database (SAD). The databases are defined to assure implementation compatibility between the deployments.

The SPD includes the static policy definition, which IP traffic requires to be protected. The policy entry gives all necessary parameters for protecting traffic: tunnel mode or transport mode, ESP or AH, encryption protocols, destination address, port and protocol of the traffic etc. The selectors are combination of: destination and source protocols, port and IP address. A policy example follows:

```

selector: destination IP address 192.168.1.0/24 protocol: any, source IP 192.168.10.0/24
ESP mode tunnel with CBC AES encryption, SHA1 for integrity protection, ESP lifetime 3600 sec
destination IP: 64.3.3.2
key delivery: isakmp
ISAKMP main mode, authentication certificates, SA life time IKE 24 hours,

```

The different implementations will have different syntax and some of the parameters can be hidden (not configurable). The SPD are pre-configured by the applications. Every SPD is linked to an outbound Security Association (SA). If no SA exists then one is created as needed. We are not discussing all parameters here, since they are deeply defined in the RFC2401 [20].

Security Association Database (SAD) contains all values relevant for an active connection. The values relevant for one connection are set in IPsec SAs. There are two types of inbound and outbound IPsec SAs, corresponding to incoming and outgoing packets. The following major parameters are included in SAD for one SA entry:

- Security Parameter Index (SPI). This is the unique identifier for the SA.
- Sequence Number is the packet number.
- Anti-Replay Window – it is the size of the anti-replay window.
- The list of packets already received within the window, usually also implemented in the SAD.
- SA lifetime is validity time from a cryptographically point of view, thus the period of time in which it can be used.
- Packet counter for all packets proceeded thought this SA. This is necessary in order to process the SA life time in bytes.
- Mode of operation can be tunnel or transport
- Protection algorithm describes the exact ESP or AH algorithm, such as 3DES_CBC etc.
- Path MTU - The observed maximum MTU, which can be used in this connection.
- Destination and source IP Address, Destination and source protocols and port
- In tunnel mode, the inner IP addresses (local IP and remote IP) protocol and port

The implementations can include supplementary parameters, such as compression algorithms, decryption errors etc. depending on their requirements

A.2.2.5 Packet processing

At the outbound processing, the packet is matched in the SPD database using the selectors. If there is a match, the packet must be protected using the parameters defined in the SA. If there is no active SA, a new SA is established. This establishment is usually done with ISAKMP negotiation see A.2.2.6. After the encapsulation, the new packet is forwarded to the routing engine.

For incoming packets, the unique SPI values are used to select an actual active SA in the SAD. The packet is decrypted according the IPsec SA. Then it is forwarded to the routing engine.

A.2.2.6 Internet Key Exchange

The IKE protocol is used for delivering the Key material to ESP or AH. It achieves also authentication and negotiation of the protection algorithms.

Before describing the protocol, let us clarify the RFC terminology. The popular Diffie-Hellman [53] algorithm gives the mathematical background for key generation in an unsecured environment. It is used in the OAKLEY [27] algorithm for key negotiation and generation with a computer working with discrete arithmetic. The IKEv1 [35] protocol is a framework defining the packet format for OAKLEY. The ISAKMP [36] is the concrete packet protocol for the negotiation of IPsec SA parameters based on the IKEv1 framework. The ISAKMP is the exact definition for use with IPsec, and IKEv1 is definition that is more abstract. The IKEv2 [34] unites the IKEv1 and ISAKMP in one protocol. For this reason, the author speaks for IKEv2 or ISAKMP based in IKEv1. In this chapter, we provide an overview of ISAKMP without presenting the mathematical backgrounds, which can be found in [53].

The ISAKMP was originally defined as a general peer-to-peer key exchange and management protocol. It was not dedicated only to ESP/AH but also other protocols. Unfortunately, it was only implemented in IPsec (ESP/AH). The ISAKMP is a high quality security protocol, divided into different phases – Phase 1 and Phase 2. The negotiation is done using UDP protocol on port 500 or 4500.

A.2.2.6.1 Phase 1 and ISAKMP SA

The first phase (Phase 1) in the ISAKMP negotiation targets establishes a secure channel for commutation. This includes mutual authentication, security algorithms negotiation, key material exchange etc. The result of phase 1 negotiation is ISAKMP Security Association (SA). The ISAKMP SA includes the parameters as authentication methods, life time of the SA, session key, called Shared Key ID, encryption methods, IDs of the participants etc. The communication in the following phase 2 is always protected by the negotiation in the phase 1 key. It is important to stress that the ISAKMP SA is not the same as IPsec SA discussed in A.2.2.4. Unfortunately, the names are very similar, but these are different entries with different scopes and targets.

The ISAKMP policy contains the parameters for the negotiation, such as authentication, encryption, lifetime, remote IP, ID etc. A general example of ISAKMP policy is shown in the following paragraph; the specific implementation can certainly include different additional parameters.

ISAKMP Policy: remote peer 63.4.2.3, encryption AES CBC or 3DES CBC, HMAC MD5, diffie-hellman group 2 or 3, lifetime max 24h, authentication pre-shared secret or RSA-Signature, remote ID FQDN, local ID IP adrs.

The main principles of the negotiation are that initiator suggests list of supported and acceptable algorithms and the responder chooses one of them. Both participants have an influence over the algorithms and values for the ISAKMP SA. ISAKMP is a peer to peer protocol, where the both participants authenticate mutually. For the authentication, the ISAKMP defines:

- *Shared secret.* Both participants share a certain secret and knowledge of the secret is sufficient for authentication. The shared secret is manually pre-configured by the administrators at the hosts. The method does not scale in large installation and has a low security level.
- *Public/Private Key.* The participants authenticate using public and private keys. This is a much better alternative than a shared secret. The main advantage is the simplicity and there is no requirement for a supplementary PKI in order to start the implementation (certificates are not used). The main shortcoming is that public key pre-distribution of the peers is needed. The initiator must have the public key of the responder in order to authenticate it and vice versa. The public keys must be known in advance, which can be difficult in environments with many participants. The certificates are not used.
- *Signatures.* The participants authenticate mutually using signatures. Certificates x509v3 [26] are involved for this purpose,. The peer signs a hash on the exchanged messages and sends it to the opposite site. The public key for verifying the signature is part of the certificate sent during the negotiation, so no pre-distribution is required. The use of certificates requires a Public Key Infrastructure (PKI). The main difficulty when using signatures

is the need for a PKI infrastructure. The main advantage is the large-scale potentials at high security level deployments. This is probably the most popular use of ISAKMP authentication.

The ISAKMP delivers a secure key for the encryption of further exchanges in phase 2. The key exchange is carried out using the famous Diffie-Hellman algorithm [53]. To facilitate the Diffie-Hellman, exchange sets of public prime numbers and elliptic curve polynomial are predefined [36]. The predefined parameters are gathered in group. There are currently four groups. The public values are exchanged during the negotiation and protected through signed hash.

There are two modes of exchange in the ISAKMP: Main Mode and Aggressive mode. The Main mode protects the ID of the peers and requires at least 6 messages. The Aggressive Mode consists of only 3 messages and does not protect the peers IDs. The exact negotiation and key derivation does not contribute directly to the mobile environment. It is described in detail in [36, 34].

A.2.2.6.2 ISAKMP Phase 2

Phase 2 negotiates the parameter for security protocol, which practically is ESP or AH. It is called Quick Mode when negotiating ESP and AH. The word Quick Mode is quasi synonym for ISAKMP Phase 2 and explained here. Additionally, there is Config Mode explained in chapter A.2.2.10.8.

The Quick Mode creates IPsec SA with the parameters required by ESP/AH protocol. The IPsec SA is unidirectional. There are two SAs – inbound and outbound for one bi-directional connection. The phase 2 negotiation is protected by the session key generated in phase 1. There can be multiple phase 2 negotiations on the top of single phase 1. Thus, many IPsec SA can be negotiated, typically one for every network segment.

A.2.2.6.3 Perfect Forward Secrecy (PFS)

The Perfect Forward Secrecy is a property of the key generation in which one key should not be a derivate of any other key. When a session key or private key is compromised at some time point, then the following (previous) session keys must not be compromised because of that. In the context of IPsec: if an attacker breaks one dedicated IPsec session key, then only this session must be compromised. The following and previous IPsec sessions remain secure. The same can be interpreted for the private key of the participants: even if the private key is compromised, then the negotiated session keys must be secure. The use of PFS is strongly recommended. IPsec achieves PFS using several mechanisms:

- There is Diffie-Hellman exchange in every quick mode negotiation, so if PFS feature is active then a KE payload is included in every Quick Mode.
- The public key is used to sign a hash value of the key material and never the key itself.
- The phase 1 must be deleted every time a phase 2 is finished in order to protect the identities.

A.2.2.6.4 Replay attacks protection

Both protocols ESP and AH deliver replay protection involving sequence numbers in the ESP/AH header. The sequence number increases monotone with every packet. The receiver checks if this sequence number has already been received and if so, it is discarded.

A window of acceptance is defined to handle the potential disorder of sequence numbers. The window defines the maximum and minimal sequence numbers, which are acceptable.

The window size is a constant number, typically 32. With every new received packet the window is moved, so the sequence number received is always at the right corner of the window. The host ensures that the packets in the window are received just once.

A.2.2.7 PMTU Discovery

The Maximum Transmission Unit (MTU) and Path MTU Discovery [52] are very important for packet processing. They define the maximum packet size on certain interfaces and certain IP paths. Some applications prefer sending single fragments and deliver the packets fast in this way. This type of applications set the DF bit flag of the IP packet, so the forwarding instances should not fragment the packet.

IPSec adds a supplementary header to the original packet. If the unencrypted packet already has the maximum size, defined as MTU, then the encrypted packet cannot be forwarded in one fragment. The packet must be encrypted, fragmented and then forwarded. If the DF bit in IP header is set then the IPSec gateway should answer the sender with ICMP notification fragmentation needed. This leads to some problems for IPSec.

Let us regard the case using IPSec between two gateways running in tunnel mode. This is a very common deployment. The IPSec gateway receives ICMP notification fragmentation needed. The gateway is only an encryption node and it must forward the notification to the end host. The ICMP packet is a reaction to certain traffic generated by the end hosts. The replay packet is sent to the gateway, because it is the sender of the protected packet. The gateway should forward the ICMP notice to the sender - end host. The end host IP is not included in the notification, since the intermediate router does not know that the packet is tunnelled etc. The gateway can only guess the originator or forward the packet to all hosts included in the IPSec SA. Both possibilities are not optimal.

The big security issue with PMTU in IPSec is that the notification sender could be an attacker. The IPSec gateway forwards an unauthenticated packet to the protected host in order to fulfil the PMTU requirements. This gives the possibility for DoS attacks on the end host. Another possibility is that the intermediate attacker can decrease the MTU size of minimal value, so that the connection becomes useless for the participants. In order to reduce these vulnerabilities the IPSec gateways using tunnel mode must inspect the ICMP packets and consider the local security policies. Blocking all ICMP "fragmentation needed" is not an option, since this ignores basic Internet principles. This can lead to disconnection see 6.2.

The simple calculation of the MTU size in IPSec connection is also problematic since the ESP header has a variable length. The length depends on the added padding and it is variable for every packet. The value of the calculation of the MTU size must consider the maximum added padding.

The PMTU [52] is generally a highly problematic feature on the Internet and not only for the IPSec. On the one hand, it is definitely needed in heterogeneous environments. On the other hand, it is a security thread and very difficult to handle. Some of the firewalls and NAT routers drop the ICMP fragmentation needed notifications. The result is that most of the implementation inclusive IPSec set the MTU size to a sufficiently low value. The value is usually an empirical derivative. This causes the unreasonable dropping of some packets and ineffective use of the network resources. The throughput of router is measured in packets per seconds and not in Bytes per seconds - better one big packet than two small.

A.2.2.8 Original implementations and ideas

There are quite different IPSec deployments described in the original RFC documents [20, 54] and the current implementations. In this chapter, the original implementation ideas are described. Later on in chapter A.2.2.11, the actual current implementations are presented.

The packets are protected according the SPD/SA policy independent, whether the peer host is currently online or not. There is no delivery proof mechanism. The higher layer protocols such as TCP must ensure the presence of the host. The protection should be part of the kernel engine, so-called “Bump-in-the-stack implementations”. This is transparent for the application and no action initiation is required. Properties such as IP, dns, winds assignment, load sharing, routing information announcement, backup are of the scope of IPSec standards. Furthermore, transparent layer 3 connectivity is assumed, thus not NAPT intermediate device.

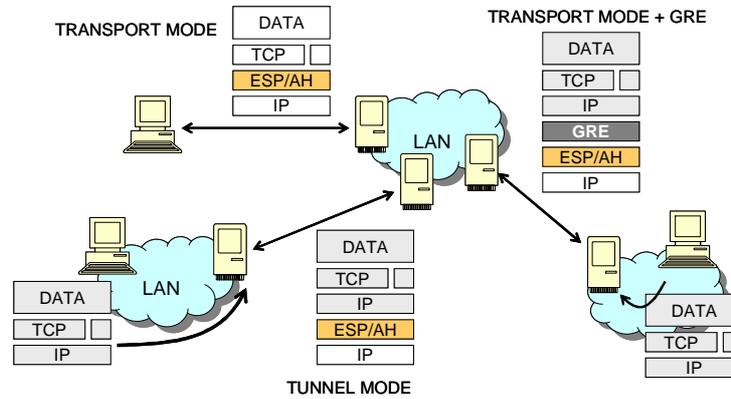


Figure 12.13: IPSec implementations

The main deployments of IPSec are shown in Figure 12.13 and defined as:

- Transport mode protection for IP communication between two hosts with fixed IP addresses.
- If the hosts do not have cryptographic possibilities, then a dedicated gateway can do this in so-called tunnel mode. The IP addresses of the host involved are also static. The connection is gateway to gateway.
- For using dynamic routing protocols, load sharing etc the implementation of GRE over IPSec in transport mode is recommended. The online status of the hosts is checked by the routing protocols. If one of the participants becomes unreachable, certain actions are taken, for example using an alternative way. The GRE is a very simple encapsulation and usually implemented as virtual interface. When the host becomes unreachable, the virtual interface can be tied down and the host can use the alternative lower metric/preferences connections.

A.2.2.9 IPSec in conjunction with NAPT

NAPT is a big issue for IPSec. This follows on from the fact that the ISAKMP and IPSec protocols try to protect the packets from manipulation by intermediate devices. NAPT is actually an intermediate device and manipulates the packet. The NAPT manipulation is an attack from a security perspective. The protocols cannot distinguish between “good” NAPT manipulation and “bad” attacker manipulation. Without going into deep sophisticated details, here are the following characteristics:

- AH protocols cannot be used with NAPT, since AH protects the IP header.
- ESP can be used in tunnel mode with NAPT devices when these support so-called “IPSec Pass through” functionality. Additionally, the ISAKMP phase

1 negotiation must not use ID payloads with IP addresses, since they are not correct.

- ESP in transport mode cannot be used since the IP manipulation causes incorrect checksum calculation in the TCP and UDP header. The TCP/UDP packets are dropped after decryption, whilst the device is verifying the checksum. For this reason, the checksum must be set to zero (do not care) by the sender according RFC [55].
- ESP transport mode is problematic. Even the end host ignores the incorrect checksum and the NAPT device supports “IPSec pass through” feature, the ESP connection in transport mode will not work. The IPSec receiver host cannot distinguish between the TCP/UDP connection and connections from different hosts behind the same NAPT device (single public IP address). The scenario is shown in Figure 12.14. Let us consider they are two different hosts: A with IP 10.0.0.1 and B with IP 10.0.0.2 in the same private LAN segment behind the same NAPT. They send ESP packets to the same destination host using the same source and destination port. After decrypting the packets, the receiver will see the same source/destination port and IP addresses for the two sessions. The receiver cannot differentiate between the two hosts. It is not possible to de-multiplex between the connections, since there is no unique parameter in the UDP/IP headers. This is a common constellation when, for example, L2TP over IPSec [33] is deployed. L2TP over IPSec is a very popular connection type, implemented in all Microsoft OS.
- ISAKMP is working, when the NAPT device supports the extended feature “multiple ISAKMP IKE session negotiations”. This feature can be part of “IPSec Pass through” set.
- ESP in tunnel mode could incur problems with NAPT. The encapsulated IP header (inner header) includes a private IP address. The private addresses can overlap, so two or more hosts can potentially have the same IPs. For example: 192.168.0.0/24 is a common address for private DSL routers. The receiver cannot differentiate between the hosts after decryption.

There are restricted cases in ESP tunnel mode, which can work in common situations. NAPT is very problematic for IPSec and for this reason the NAT-T feature was developed. It is described in chapter A.2.2.10.1.

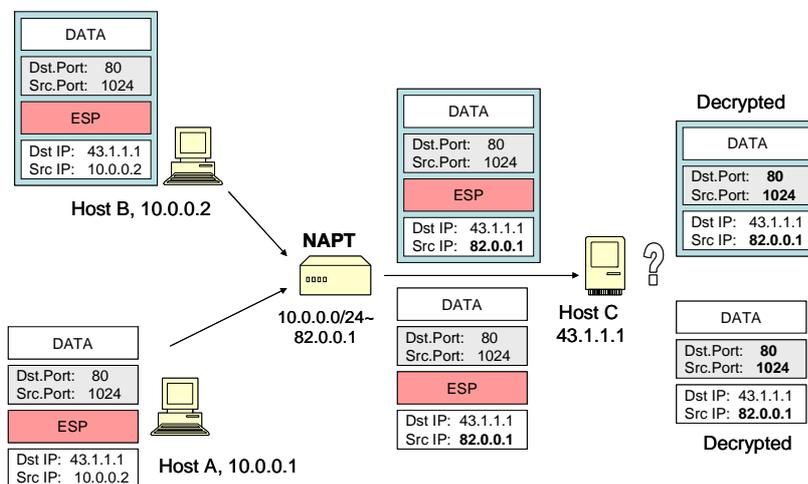


Figure 12.14: Transport mode of ESP used with NAPT

A.2.2.10 Enhanced IPsec features

In the recent years, many new features of ESP/ISAKMP have been developed, which have responded to the new requirements of industry. The motivation for the features was NAT popularity and remote corporate access. The features were originally proprietary solutions, later standardised in IETF. This chapter provides a short overview of the new features and then the current IPsec usages are presented.

A.2.2.10.1 NAT-Traversal (NAT-T)

The popularity of NAT requires solution so that IPsec can be used in these environments. A NAT-Traversal was developed for ISAKMP and IPsec was [29, 28, 30]. It extends the protocols with the following features:

- Multiple ISAKMP SAa (phase 1) can be established between the same hosts. The NAT device represents to multiple private hosts. The original RFC restricts to only one session between two hosts.
- Source port ISAKMP negotiation can be any port. Originally it was only UDP 500
- The NAT-T support is signalled using vendor-ID payloads. If an NAT device is detected, the transport port is changed from UDP 500 to 4500.
- The ESP and AH packets are encapsulated in UDP. The NAT device can thread the ESP/AH packet as a standard UDP packet. The UDP header has the destination port 4500. The layer structure is shown in Figure 12.15. Through decryption, the UDP/TCP checksum of the inner header is recalculated or ignored.
- Messages, called “nat-t-keep-alive”, are used to keep the NAT binding active. As described in chapter A.2.1, the NAT translation is deleted after a certain period of inactivity (idle timeout). In order to keep the translation active, the NAT-T implements keep-alive-updates. This is implemented as part of ISAKMP phase 1 with empty messages called “nat-t-keep-alive”. If there is an NAT, at least one of the peers must send packets regularly in order the keep the NAT entry.
- ISAKMP SA session is active during the whole communication and requires some CPU resources to precede the keep-alive messages.

In most of the cases, NAT-T will successfully assist connection establishment. The NAT-T feature cannot handle all NAT cases, such as UDP fragmentation cases for example. In ISAKMP, the participants can use certificates for authentication. A certificate is usually larger than 1500 bytes and the packets will be fragmented. The fragmented UDP packet can be dropped by the NAT device upon which the ISAKMP will fail, see chapter A.2.1.9

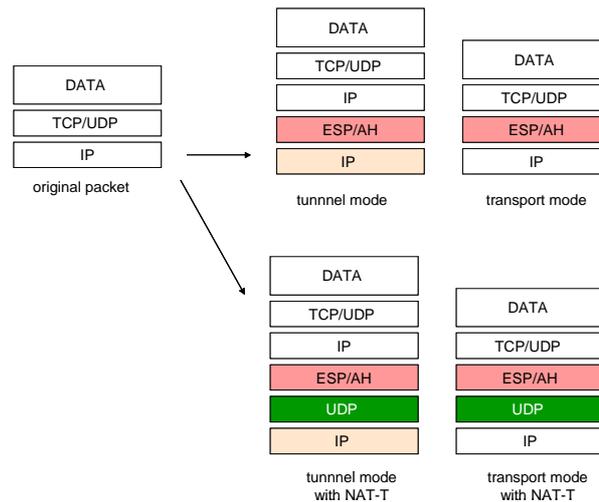


Figure 12.15: ESP/AH layer and NAT-T

A.2.2.10.2 Dead Peer Detection

IPSec is IP layer security protocol. The IP protocol does not handle host status information, i.e. whether it is reachable or not. Originally, IPSec was designed as a connectionless protocol - the receiver can be online or offline. With the spreading of the tunnel mode, a new feature known as Dead-Peer-Detection (DPD) [57] was introduced. This feature detects whether the peer is reachable. It is based on ISKMP phase 1 messages. The DPD feature support is announced using the vendor ID payloads. The receiver of the DPD message must reply and in this way assure its presence. The IPSec SA is deleted if there is no reply to the DPD request.

A.2.2.10.3 Reverse Route Injection

The IDs in ISAKMP Quick mode of are commonly used to announce an IP address or networks to the opposite peer. The common example is the use of ESP tunnel mode, where the IP network parameters of the initiator are not known to the responder in advance. The responder serves many initiators with unknown public IP addresses. In order to set a proper routing entry, the initiators IP address must be extracted from the ISAKMP negotiation and added to the routing table. This is called “reverse route injection”. The feature sets the quick mode ID as a directly connected network in the routing table of the gateway. The route entry is usually redistributed in the backbone.

A.2.2.10.4 Idle Timeout

If any IPSec packets are received at certain times then the session is deleted. This new feature was introduced by all vendors, because the duration of the communication was used for billing. The customers are paying on a time basis for using the IPSec resources. Another motivation is to reduce the “dead/zombie” sessions and keep the memory free.

A.2.2.10.5 ISAKMP(IKE) Pass through

A solution to the NAPT problem was also developed by the device vendors. It was created in parallel to the NAT-T described in A.2.2.10.1, so both of them currently co-exist. The target was to solve the problem on the intermediate NAPT device, without changing the original IPSec standard. When an NAPT router is between the hosts during ISAKMP

negotiation it requires some mechanism to map the return ISAKMP messages to the correct inside host. If two hosts behind the NAT device negotiate with the same destination peer, the returned packets have the same source, destination port and IP address. De-multiplexing at the NAT device was not possible, since there is no unique parameter. The IKE pass-through uses the cookie in the ISAKMP header to map the packets in the right connection.

A.2.2.10.6 IPsec Pass through

The NAT vendor solution for the ESP header was developed and called “IPsec Pass through”, similar to “IKE Pass through”. The idea was to find a unique identifier for de-multiplexing the return packet. The ESP header includes the SPI value, which is unique and can be used for this purpose. The problem is that the SPI number is negotiated in the ISAKMP quick mode phase, which is encrypted and inaccessible to the NAT device. The solution is based on the assumption that send-received packets are coming in very short intervals after the ISAKMP establishment. The NAT router waits for input and output ESP traffic and creates the mapping on the fly during the first ESP packets. There are inbound and outbound SPI, so traffic is needed in both directions. A critical point of this algorithm is that it does not always work. The parallel establishment of multiple connections means that it is not possible to pick up the correct SPI. In a small office this feature will assist connection establishment, but it will not be helpful in large-scale environments.

A.2.2.10.7 Not interrupted mode

A new IPsec SA must be renegotiated before the session lifetime expires. An overlap of two IPsec SA with the same SPD is permitted, in order to ensure that packets are not dropped during the negotiation of the new session. The old and new IPsec SA can coexist and handle packets encrypted with the new and old key. No packet losses are achieved during the reestablishment.

A.2.2.10.8 Phase 2 Conf mode (Easy VPN)

The idea of Easy VPN was to use ESP in tunnel mode for client to server communication. The VPN gateway (server) assigns a private IP address and DNS/WINS to the VPN client. A draft called Conf Mode [32] was specified to enable this. The IPsec client negotiates phase 1 with its public IP address. The client starts then Conf Mode and receives the corporate IP address/parameter etc. After it, the Quick mode is established with the new private IP parameter. The Conf mode is an expired draft, which has never turned in RFC. Although it has not standardised, it is used in most of the IPsec client software. In practical terms, the biggest installations of IPsec for remote workers use this draft, like Cisco VPN.

A.2.2.10.9 Extended Authentication (X-Auth)

The X-Auth [31] gives the option of user authentication with username/password supplementary to the phase 1 authentication. There are two authentications, one in phase 1 and one in X-Auth with username and password. Typically, the user authenticates himself using username/password and the computer with a certificate. This is also an expired draft, which, because of its controversy, never became RFC. The X-Auth is very popular in the IPsec software for remote workers, like Cisco VPN.

A.2.2.11 Practical IPsec deployments

The IPSec deployments have changed a lot from the original ideas described in chapter A.2.2.8. There are two main deployments (Figure 12.16): corporate access to remote workers (single PC) and the connection of small offices.

There are two main alternatives for the remote access: (1) tunnel mode with support of NAT-T, X-Auth, Config mode. Cisco VPN client is one of the leading representatives. (2) The modern clients are using “L2TP over IPSEC” with NAT-T support [33] (Figure 12.16). The main deployment is the Microsoft VPN Client part of all Microsoft OS (Mobile, 2000, 2003, XP, Vista).

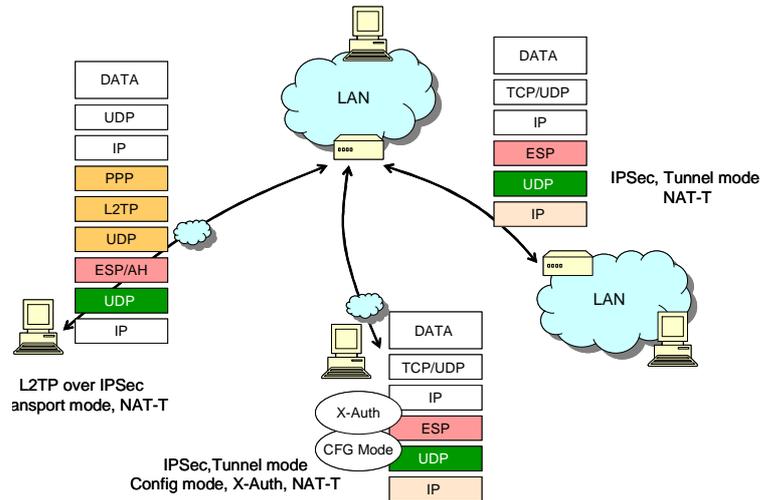


Figure 12.16: Current IPSec deployments

The second usage of IPSec is for connecting small LANs to a main office through the Internet, where tunnel mode with NAT-T is used see Figure 12.16.

A.3 References in the appendix A

- [1] Egevang, K., and P. Francis, "The IP Network Address Translator (NAT)," RFC 1631, May 1994.
- [2] Srisuresh, P., and D. Gan, "Load Sharing Using IP Network Address Translation (LSNAT)," RFC 2391, August 1998.
- [3] Srisuresh, P., and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations," RFC 2663, August 1999.
- [4] Tsirtsis, G., and P. Srisuresh, "Network Address Translation—Protocol Translation (NAT-PT)," RFC 2776, February 2000.
- [5] Hain, T., "Architectural Implications of NAT," RFC 2993, November 2000.
- [6] Srisuresh, P., and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)," RFC 3022, January 2001.
- [7] Holdrege, M., and P. Srisuresh, "Protocol Complications with the IP Network Address Translator," RFC 3027, January 2001.
- [8] D. Senie, "Network Address Translator (NAT)-Friendly Application Design Guidelines," RFC 3235, January 2002.
- [9] Srisuresh, P., J. Kuthan, J. Rosenberg, A. Molitor, and A. Rayhan, "Middlebox Communication Architecture and Framework," RFC 3303, August 2002.
- [10] Daigle, L., and IAB, "IAB Considerations for Unilateral Self-Address Fixing (UNSAF) Across Network Address Translation," RFC 3424, November 2002.
- [11] Rosenberg, J., Weinberger, J., Huitema, C., and R. Mahy, "STUN—Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)," RFC 3489, March 2003.
- [12] Aboba, B., and W. Dixon, "IPsec—Network Address Translation (NAT) Compatibility Requirements," RFC 3715, March 2004.
- [13] Audet, F., and C. Jennings, "NAT/Firewall Behavioral Requirements," work in progress, draft-audet-nat-behave, July 2004.
- [14] Ford, B., P. Srisuresh, and D. Kegel, "Peer-to-Peer(P2P) Communication across Network Address Translators (NATs)," work in progress, draft-ford-midcom-p2p, June 2004.
- [15] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Methodology for Network Address Translator (NAT) Traversal for the Session Initiation Protocol (SIP)," work in progress, draft-ietf-mmusic-ice, July 2004.
- [16] Jennings, C., "NAT Classification Results Using STUN," work in progress, draft-jennings-midcom-stun-results, July 2004.
- [17] Rosenberg, J. Weinberger, R. Mahy, and C. Huitema, "Traversal Using Relay NAT (TURN)," work in progress, draft-rosenberg-midcom-turn-01, July 2004.
- [18] Diffie, W., M. Wiener, P. Van Oorschot, "Authentication and Authenticated Key Exchanges, Designs, Codes, and Cryptography", 2, 107-125, Kluwer Academic Publishers, 1992.
- [19] Kent, S., "IP Authentication Header", RFC 4302, December 2005.
- [20] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.
- [21] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.
- [22] Pereira, R., and R. Adams, "The ESP CBC-Mode Cipher Algorithms", RFC 2451, November 1998.
- [23] Glenn, R., and S. Kent, "The NULL Encryption Algorithm and Its Use With IPsec", RFC 2410, November 1998.
- [24] Madson, C., and N. Doraswamy, "The ESP DES-CBC Cipher Algorithm With Explicit IV", RFC 2405, November 1998.
- [25] Madson, C., and R. Glenn, "The Use of HMAC-MD5 within ESP and AH", RFC 2403, November 1998.
- [26] ISO/IEC 9594-8, "Information Technology - Open Systems Interconnection - The Directory: Authentication Framework", CCITT/ITU Recommendation X.509, 1993.
- [27] Orman, H., "The OAKLEY Key Determination Protocol", RFC 2412, November 1998.
- [28] Kivinen, T., "Negotiation of NAT-Traversal in the IKE", RFC 3947, January 2005.
- [29] Aboba, B. and W. Dixon, "IPsec-Network Address Translation (NAT) Compatibility Requirements", RFC 3715, March 2004.
- [30] Huttunen, A., Swander, B., Volpe, V., DiBurro, L., and M. Stenberg, "UDP Encapsulation of IPsec Packets", RFC 3948, January 2005.
- [31] R. Pereira et al., "Extended Authentication within ISAKMP/Oakley (XAUTH)", 2000, IETF, draft-ietf-ipsec-isakmp-xauth-06.txt (May 2000 expiration)
- [32] R. Pereira et al., "The ISAKMP Configuration Method", 1999 IETF, draft-ietf-ipsec-isakmp-mode-cfg-05.txt
- [33] Patel, et al., "Securing L2TP using IPsec", IETF, November 2001
- [34] Kaufman, C., Ed., "The Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005
- [35] Harkins, D., and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.
- [36] Maughan, D., Schertler, M., Schneider, M., and J. Turner, "Internet Security Association and Key Management Protocol (ISAKMP)", RFC 2408, November 1998.
- [37] Kent, S. et al. "Security Architecture for the Internet Protocol", RFC 2401, November 1998
- [38] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.
- [39] 3GPP TS 22.101: "Universal Mobile Telecommunications System (UMTS): Service aspects; Service principles"
- [40] 3GPP TS 23.101: "Universal Mobile Telecommunications System (UMTS): General UMTS Architecture"
- [41] Internet Engineering Task Force www.ietf.org
- [42] Gast, Matthew S., "802.11 Wireless Networks / The Definitive Guide. (Definitive Guide)", O'Reilly Media, 2006
- [43] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.

- [44] IEEE, "IEEE std 802.3 -2005/Cor 1-2006 IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks", IEEE Std 802.3-2005/Cor 1-2006 (Corrigendum to IEEE Std 802.3-2005)
- [45] IEEE, "IEEE standard for local and metropolitan area networks virtual bridged local area networks", IEEE Std 802.1Q-2005 (Incorporates IEEE Std 802.1Q1998, IEEE Std 802.1u-2001, IEEE Std 802.1v-2001, and IEEE Std 802.1s- 2002)
- [46] Santitoro, Ralph, "Metro Ethernet Services – A Technical Overview", Metro Ethernet Forum, 2006
- [47] ATM Forum, www.atmform.org
- [48] Phishing Form, <http://www.antiphishing.org/>
- [49] Postel, J., and J. Reynolds, "File Transfer Protocol", STD 1, RFC 959, USC/Information Sciences Institute, October 1985.
- [50] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002
- [51] Postel, J., "Internet Control Message Protocol", RFC 792, STD 5, September 1981.
- [52] Mogul, J., and S. Deering, S., "Path MTU Discovery", RFC 1191, DECWRL, Stanford University, November 1990
- [53] Stallings, William, "Cryptography and Network Security", Prentice Hall, November 2005
- [54] Bollapragada, Vijay, Khalid, Mohamed, Wainner, Scott, "Advanced IPSec VPN Design", acmillan Technical Publishing, April 2005
- [55] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.
- [56] Santiago Alvarez, "QoS for IP/MPLS Networks", Macmillan Technical Publishing, Juni 2006
- [57] Huang, et al., " A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers", RFC 3706, February 2004