SECURE AND SEAMLESS PAYMENT FOR WIRELESS MESH NETWORKS

by

SERHAT CAN LELOĞLU

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabanci University
January 2013

SECURE AND SEAMLESS PAYMENT FOR WIRELESS MESH NETWORKS

APPROVED BY

Assoc. Prof. Dr. Albert Levi .....

(Thesis Supervisor)

Asst. Prof. Dr. Cemal Yılmaz .....

Assoc. Prof. Dr. Erkay Savaş .....

Assoc. Prof. Dr. Özgür Erçetin .....

Assoc. Prof. Dr. Yücel Saygın .....

DATE OF APPROVAL .....

SECURE AND SEAMLESS PRE-PAYMENT FOR WIRELESS MESH NETWORKS

Serhat Can Leloğlu
Computer Science and Engineering, MS Thesis, 2013
Thesis Supervisor: Assoc. Prof. Albert Levi
Keywords: Prepaid Payment Systems, Security, Micropayments, Wireless Mesh Networks

Abstract

Wireless Mesh Network (WMN) is a multi-hop high-speed networking technology for broadband network access. Compared to conventional network service providing systems such as base stations, WMNs are easy to deploy and cost-effective systems. We propose a secure and seamless pre-payment system for Internet access through WMNs. The proposed system is called and will be mentioned as SSPayWMN. The system will be fair to both clients and to service providers. Since service providers intentionally or unintentionally may overcharge the clients, SSPayWMN offers cryptographic proofs for given Internet service.

Related works for broadband access typically have full trust to operators; however in real life operators may unintentionally overcharge their users. This misbehavior in the system may cause disputes between the clients and the operators. Even when the operator is right, it is very difficult to convince the customer since the operators generally do not have justifiable proofs that cannot be denied by the clients.

The proposed system's main goal is to providing a secure payment scheme, which is fair to both the operator and the clients. Using cryptographic tools and techniques, all system entities will be able to authenticate each other and provide/get service in an undeniable way. Implementing the system on a network simulator proves the proposed system's effectiveness. Network simulation results ensure real life performance results for critical use cases.

Untraceability is a system property, which unable an adversary to trace actions of a client. SSPayWMN protects clients' anonymity and provides untraceability for clients. The implementation of the system is made on a network simulator and simulation results are presented. SSPayWMN has achieved remarkable results in the simulations; system protocols reached steady state in every simulation, which ensures the stability of the system.

KABLOSUZ ÖRGÜ AĞLARI İÇİN GÜVENLİ VE KESİNTİSİZ ÖN ÖDEMELİ SİSTEM

Serhat Can Leloğlu
Bilgisayar Bilimi ve Mühendisliği, Yüksek Lisans Tezi, 2013
Tez Danışmanı: Doç. Dr. Albert Levi
Anahtar Kelimeler: Ön Ödemeli Sistemler, Güvenlik, Mikro Ödeme, Kablosuz Örgü Ağlar

Özet

Kablosuz Örgü Ağları (KÖA) geniş alanda erişilebilir, ağ paketlerini çok zıplamalı şekilde hedefe ulaştırabilen, yüksek bağlantı hızı sağlayabilen bir ağ teknolojisidir. Alışılagelmiş yollara nazaran kurulumu daha kolaydır ve daha ekonomiktir. KÖA geniş alanlarda hareketli ve hareketsiz kullanıcılara bağlantı sağlayabildiği için ağ hizmeti sağlayıcı sistemler için uygun teknolojilerdir. KÖA ağlarındaki elemanlar ağ içinde bir elemana paket gönderdiklerinde kullandıkları rotayı hafızalarında tutarlar. Bu da onlara bir sonraki paket teslimatında daha hızlı olmalarını sağlar. Rota tabloları isteğe göre önden de yüklenebilir.

Önerilen sistemde güvenilir bir üçüncü parti (GÜP) görev alıyor. GÜP'ün yanı sıra kullanıcılar ve servis sağlayıcı operatörler de mevcut. GÜP aktif olarak sistemin içerisinde yer alıyor ve operatörler servis sağlanacak olan bölgeye belli aralıklarla erişim noktaları koyacaklar. Kullanıcılar sinyal gücü en yüksek olan erişim noktasını tercih edecekler bu da operatörler arasında daha iyi servis sağlamak için bir rekabete sebep olacak. Kullanıcılar belli bir operatörün sabit müşterileri olmayacaklar. Sinyal gücünde değişiklik olursa veya daha güçlü bir ulaşım noktasının yakınından geçiliyorsa kullanıcı hizmet aldığı ulaşım noktasını değiştirebilir. Bu değişiklik kullanıcıya servis bekleme süresinde artış veya bağlantı kesintisi şeklinde yansımayacak.

Sistemin ana amacı operatörlere mutlak güven ilkesinin benimsenmediği durumları kapsayacak bir ödeme yolu sağlamak. Önerilen sistem operatörlerin de bilinçli veya bilinçsiz şekilde fazladan para almasını engelleyecek. Sağlanan hizmet kriptografik yollarla kanıtlanabilecek bunun yanı sıra sağlanmayan hizmet için kanıt sunulamazsa bu hizmetin hiç sağlanmadığı anlaşılacak. Sistemin doğru ve efektif bir şekilde çalıştığını gösterebilmek için ağ simülasyonları da yapıldı. Gerçek hayata daha yakın sonuçlar elde edebilmek için kullanıcı tipleri düşünüldü. Simülasyonlar bu kullanıcı tiplerini de katarak yapıldı.

To my dear family

ACKNOWLEDGEMENTS

I would like to thank my thesis supervisor, Albert Levi, who has offered me this project at the beginning of my master studies. He has always been very helpful and understanding.

I would like to thank Yücel Saygın for introducing a new area of computer science to me with Information Retrieval and Data Mining courses. He was very kind by accepting to join jury.

I would like to thank Erkay Savaş for giving me my cryptographic basis in my undergraduate studies. I would like to thank him for being a part of my jury.

I also thank Özgür Erçetin and Cemal Yılmaz for devoting their time for jury during their tight schedule.

I would like to thank my wife and family for their support and motivation.

Finally, I thank Türk Telekom for funding my project and scholarship.

Contents

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Wireless Mesh Networks [1] offer broadband network access with high-speed network connection. WMNs are easy to deploy and cost effective compared to conventional Internet service providing infrastructures such as high-powered servers. Mesh networks dynamically organize themselves and they do not need a centralized element, in that sense they are a subset of ad-hoc networks. Mesh nodes deliver packets from source to destination in a multi-hop manner, conclusively they extent network coverage. WMNs could support for both mesh purposes and also conventional Wi-Fi connections. WiMax [4], ZigBee [5] and 3G-radio access [29] could also inter-connect with WMN structure.

SSPayWMN employs some cryptographic primitives to ensure system security. The billing system counts on hash chains [32] and uses every element of the hash chain as a token, which buys time intervals with Internet service. SSPayWMN employs a Trusted Third Party (TTP), who ensures honest usage of the system by every party. The packets that are transmitted are either encrypted or transmitted on a secure line.

SSPayWMN is designed to reckon with real-life challenges such as stable Internet service during client mobility and rush hours. To estimate SSPayWMN performance, network simulations for the proposed system are executed. The simulations are divided into two groups. The former is unit tests, which simulate a unit of the system and check if it is fit to use. A unit in SSPayWMN corresponds to network protocols. The latter simulation group is called real-life scenario simulations. In these simulations the clients are selected considering human behavior and they are grouped into different groups. Unit simulations provided considerable results and in all of the simulations SSPayWMN reached steady state performance. In real-life scenario simulation results the system reached steady state also, which ensures system stability.

1.1 Contribution of the Thesis

There has been research for developing secure pre-payment systems for Internet access. In [30], the authors use a high-level approach for billing and propose architecture. Their focus is mostly its performance on a threshold based bandwidth management algorithm. In [31], the authors propose UPASS; a double hash chain based prepaid billing architecture for WMNs. Their trust model is based on both classical certificate-based public-key cryptography and

identity-based cryptography. The drawbacks of [30] are the complex trust and payment structures, missing simulative and/or analytical performance model, and disregarding users' anonymity/privacy. Similarly, UPASS does not consider client anonymity and untraceability. The proposed secure and seamless system will implement a prepaid billing scheme with simpler structures and trust models. Authentication, user and operator non-repudiation, settlement and especially user privacy is taken into consideration in the system design.

SSPayWMN is designed to reckon with real-life challenges such as stable Internet service during client mobility and rush hours. To estimate SSPayWMN performance, network simulations for the proposed system are executed. The simulations are divided into two groups. The former is unit tests, which simulate a unit of the system and check if it is fit to use. A unit in SSPayWMN corresponds to network protocols. The latter simulation group is called real-life scenario simulations. In these simulations the clients are selected considering human behavior and they are grouped into different groups. Unit simulations provided considerable results and in all of the simulations SSPayWMN reached steady state performance. In real-life scenario simulation results the system reached steady state also, which ensures system stability.

Authentication, confidentiality, non-repudiation, fraud protection is provided in the system. The users will not be able to deny using credits for the services actually obtained; the operator will not be able to charge more than the usage amount. Additionally, inter-operator settlement will be performed in a secure way such that each operator will have cryptographic proofs of use for the services that they provide to other operators' customers. In order to provide privacy of individuals, our scheme will provide untraceability such that no unauthorized entity will be able to track down a particular user.

Since the clients are mobile, they may hand over among different mesh routers (i.e. access points) of the same operators. They may also roam among different operators, not only due to coverage reasons, but also for having a better quality service. Our system aims to have seamless mobility and seamless roaming for payment purposes such that when the client gets service through a new AP or switch to another operator, authentication and authorization are not performed from scratch.

From security point of view, we aim to have mutual authentication between client and the network in our protocols. Anonymity of the clients and untraceability across different

usage periods (a.k.a. untraceability) are privacy related goals of the protocols.

From payment point of view, our main aim is to have a fair system in which all the claimed transactions bear cryptographic proofs. In this way, the clients cannot repudiate using a service and the operators cannot claim for services that they do not provide. The latter is especially important during inter-operator settlement; it is also important to resolve client disputes.

1.2 Organization of the Thesis

The organization of the thesis is as follows. Brief background information is given in Section 2. Cryptographic primitives and algorithms are explained in Section 3. Requirements for a secure and seamless pre-payment system are described in Section 4. Section 5 discusses the evolution of SSPayWMN. In section 6 the designed protocols for the proposed system are presented. In section 7 the settlement of the operators and money exchange system are explained. In Section 8 there is a discussion about the success of the proposed system on meeting the previously explained system requirements. Unit test results are located in Section 9. Client types and mobility are described in Section 10. Real life scenario simulation results are placed under Section 11. Conclusively, Section 12 gives the conclusion.

2. BACKGROUND ON WIRELESS MESH NETWORKS

Wireless Mesh Network (WMN) is multi-hop wireless networking type, designed as an alternative to traditional centralized wireless networking achieved by mesh routers [1]. Mesh routers and mesh network clients form up WMNs. Each mesh node functions as a host and also as a router, relaying packets on behalf of other nodes, connecting nodes that are not located within the transmission range of each other. WMNs create ad-hoc networks, which are dynamically self-organized and self-configured. WMNs are easy to deploy and cost-effective systems, they are easy to maintain and provide robustness and reliable service coverage.

WMNs comprise of two types of nodes: mesh routers and mesh clients. A wireless mesh router provides mesh networking by using routing functions that do not exist in common wireless routers with gateway/repeater capabilities. Mesh routers have multiple wireless interfaces to expand flexibility of WMNs. Mesh routers in WMNs achieve wider coverage compared with conventional routers by using multi-hop technology with lower transmission power. Moreover it is possible to postulate improved scalability by optimizing the medium access control (MAC) protocol in a mesh router.

2.1 Network Architecture

Three main groups depending on operation of the nodes could accomplish the categorization of WMNs.

Infrastructure/Backbone WMNs: The architecture is shown in Figure 2.1. Both wireless and wired networks comprise infrastructure WMNs, in Figure 2.1 dash lines depict wireless connections whereas solid lines depict wired communications. Mesh routers establish an infrastructure to mesh clients to connect. The infrastructure is a cloud from the clients' point of view. It is a black box that delivers packets originated from the clients to the gateways.

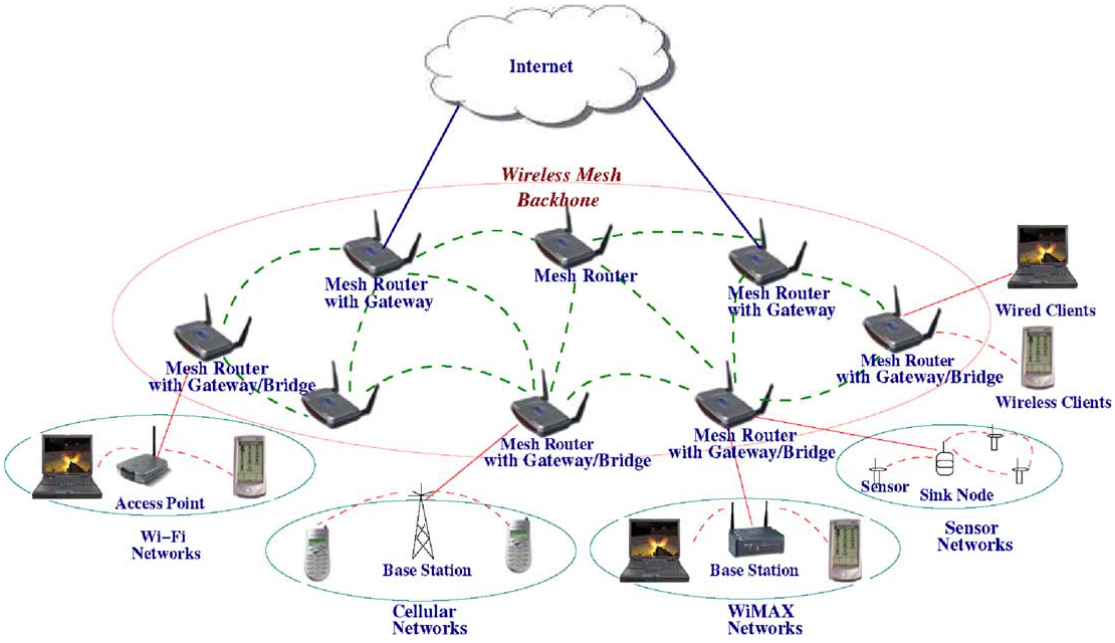

Figure 2.1. Infrastructure/Backbone WMNs. [1]

The mesh routers are self-configured and self-healing. In a case of node addition or removal, mesh backbone configures itself by forming up neighborhood. Additionally, mesh routers could connect to the Internet with gateway functionality. Infrastructure meshing provides easy to access to Internet by forming up clouds for clients. Bridging and inter-networking functionalities of WMNs enable clients to connect to mesh backbone with conventional Wi-Fi or cellular devices and also via Ethernet links. As depicted in Figure 2.1 base stations could also connect to mesh backbones, which provide Internet connectivity for all the clients of base stations.

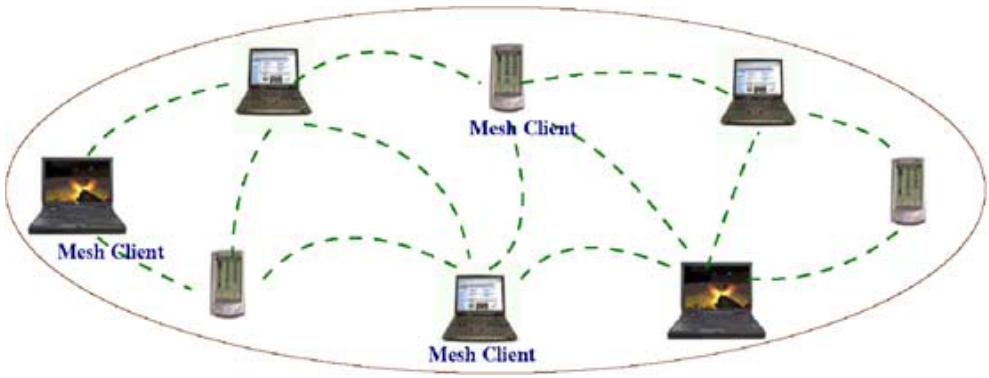

Figure 2.2. Client WMNs [1]

Client WMNs: Client meshing is a subset of Infrastructure meshing. As previously explained mesh routers establish a backbone for mesh clients, however in

client meshing case the whole network is a backbone and whoever wants to join to the network has to be a part of the backbone and provide routing functionality. As shown in Figure 2.2 client meshing is a commune type of networking.

Hybrid WMNs: This architecture is combination of two previously explained mesh architectures. Mesh clients can access Internet through mesh backbone whereas they can communicate within each other by using a simple ad-hoc network.

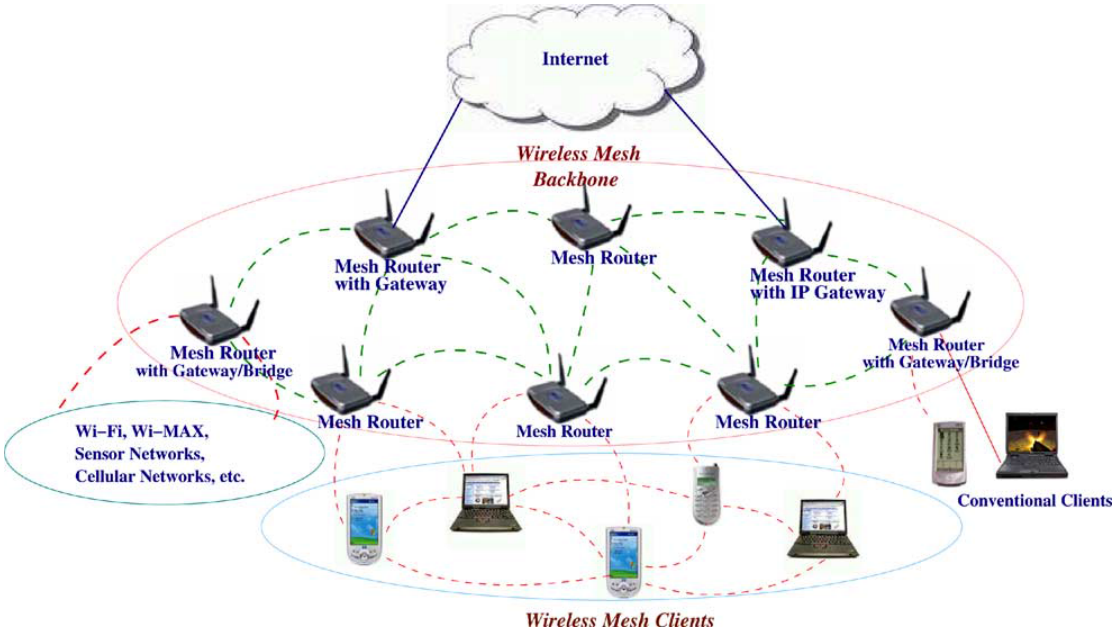

Figure 2.3. Hybrid WMNs [1]

As shown in Figure 2.3 mesh backbone provides Internet connectivity whereas Client WMNs provide connectivity to mesh backbone for far located mesh clients.

2.2 Characteristics of Wireless Mesh Networks

Characteristics of WMNs are explained as follows:

Multi-hop Wireless Network: Main accomplishment of WMNs is providing extended wireless network coverage without increasing transmission power or additional antennas.

Support for Ad-hoc Networking: WMNs provide flexible networking, which has the abilities like self-configuring and self-healing. Deployment, node addition and removal are easy to accomplish since mesh routers form routing paths by themselves.

Mobile Dependence on the Type of Mesh Nodes: Mesh routers usually do not change their locations, whereas mesh clients are assumed to be mobile.

Multiple Types of Network Access: Mesh routers are accessible via IEEE 802.11 protocols and also peer-to-peer protocols.

Dependence of Power-Consumption Constraints on the Type of Mesh Nodes: Mesh routers do not have power-consumption constraints in common but it is advisable for mesh clients to have some forms of power consumption constraints.

Compatibility and Interoperability with Existing Wireless Networks: WMNs are compatible with IEEE 802.11 protocols [2, 3], therefore WMNs could support for both mesh purposes and also conventional Wi-Fi connections. WiMax [4], ZigBee [5] and 3G-radio access [29] could also inter-connect with WMN structure.

3. BACKGROUND ON CRYPTOGRAPHIC ALGORITHMS

To establish a secure system, cryptographic primitive algorithms are employed. A brief explanation and introduction for cryptographic primitives are provided in this thesis to provide unity in the document.

In following sections hash functions, hash chains and HMAC functions are explained. Moreover symmetric cryptography is described. Finally explanation for public key cryptography is provided at the end of this section.

3.1 Hash Functions

Hash functions [7] are irreversible mathematical functions that map input strings of variable length to fixed sized output strings. Hash functions are usually employed for improving time performance of table lookup or data comparison tasks such as finding items in a database, discovering repeated or analogous records in a bulky file, finding similar springs in a DNA string and cryptographic purposes.

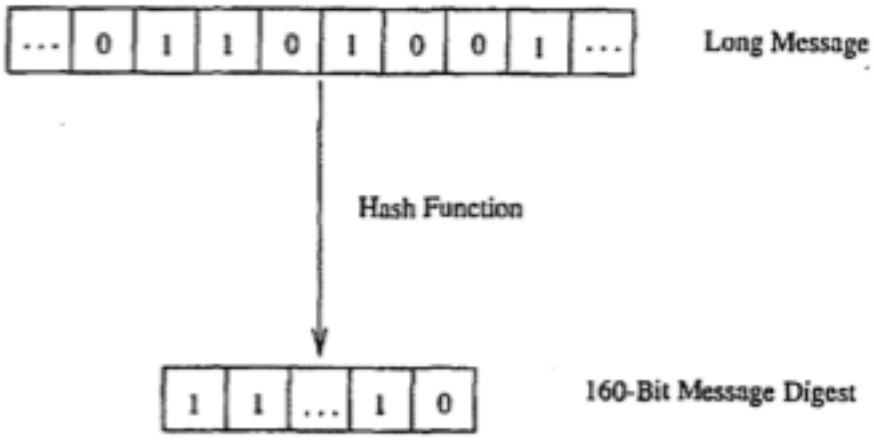

Figure 3.1. Hash Function Example [7]

Figure 3.1 depicts the hash function flow. The function maps a longer message to a 160-bit bit string. The output length depends on the hash algorithm; various hash algorithms have different output sizes.

Hash functions could receive various sized parameters but generate fixed sized input strings. Compared to mainstream cryptographic algorithms, hash functions are fairly cost-effective in both power and time consumption. Light-weightiness of hash functions make them eligible for security systems.

A hash function should satisfy following properties:

1. Given a message m, the message digest h (m) can be calculated very quickly.
2. Given a y, it is computationally infeasible to find a relation with h (m'). $=Y$ (in other words, h is a one-way, or collision resistant, function)

The most popular and well-regarded hash functions are MD5 [24], SHA1 [23] and SHA2 [23], which is a set of SHA-224, SHA-256, SHA-384 and SHA-512.

3.2 Hash Chains

Applying a hash algorithm to an initial value and using the output as an input for the next hash function forms a hash chain. Every output of a hash algorithm represents a link in the chain. Length of the hash chain [32] is determined by the number of times the hash algorithm is executed.

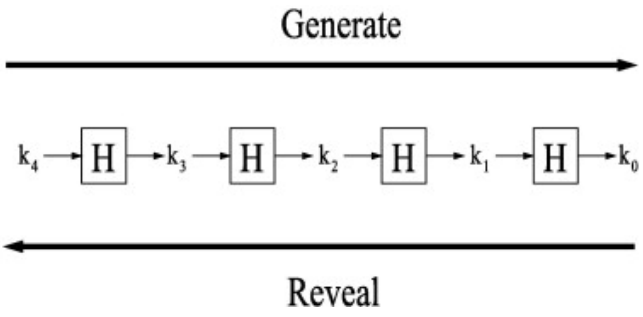

Figure 3.2. Hash Chain Depiction and Usage [8]

Since hash functions are irreversible, as shown in Figure 3.2 it is easy to go forward in the chain but it is not computationally feasible to go backwards. Which means a person could find any value on the chain if she knows the initial value but this situation is not possible for a person who knows the last value on the chain.

A hash chain with n elements is denoted as:

$$\underbrace{h\Big(h\big(h(\dots h(x)\dots)\big)\Big)}_{\text{n times}} = h^n(x)$$

Because of the fact that hash functions are one-way mathematical functions, it is appropriate to say that hash functions are good tools for security systems communicating through insecure links. Knowing the first link in the chain gives the opportunity to verify the following links in the chain as well. If one could establish a system, successful at distributing the first link in the chain, it is feasible to use hash chains as future keys or secrets for other cryptographic functions.

Hash chains are easy-to-deploy and cost-effective therefore they are widely used in cryptographic systems. Especially for systems that have delicacy for computational delay hash chains are effective tools.

3.3 HMAC Functions

One of the main research areas in cryptography and computer networking is providing integrity and reliability on a transmitted or stored data. Classically, MACs are used between two parties that share a secret key in order to authenticate the transmitted or stored data between these parties. This protocol executes a MAC that uses a cryptographic hash function union with a secret key.

HMACs are used together with widely accepted hash functions. HMAC employs a secret key for generation and verification of the MACs. The aims of HMAC construction [9] are:

- Using hash functions without making any changes on them. Previously implemented codes and hardware shall work with the deployment of HMAC.
- Maintaining the original fastness of the hash functions.
- Using and handling secret keys in a cost-effective way.
- Providing provable and reasonable cryptographic analyzes using the previously done performance analysis of underlying hash functions.
- Achieving faster and more robust performances in a case of a faster hash function is invented in the future. Replacement should be easy-to-achieve.

Table 3.1 explains the parameters HMAC uses.

Table 3.1. HMAC Parameters [9]

B	Block size in bytes
H	A secure and fast hash function
ipad	Inner pad, the byte x36 B times
K	Shared secret key
K_0	The key K before any process to make it B bytes long
L	Block size of the output of the hash function, in bytes
opad	Outer pad, the byte x5c repeated B times
t	The number of bytes of MAC.
text	The data that used to calculate HMAC
xN	Hexadecimal notation where each string N represents 4 binary bits
$\oplus$	Exclusive-Or operation
$\|\|$	Concatenation

Figure 3.3 shows the steps of HMAC.

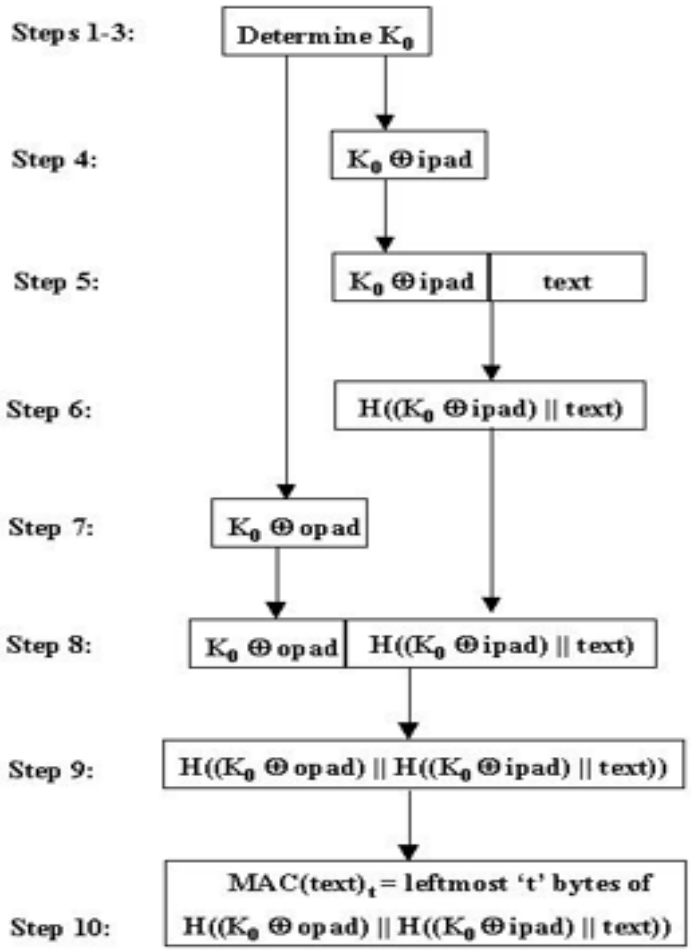

Figure 3.3. Steps of HMAC [9]

3.4 Symmetric Cryptography

Symmetric cryptography is the oldest kind of cryptographic primitive. This primitive employs shared secret keys between two parties. The security level of a symmetric cryptographic algorithm mostly depends on key size. Modern algorithms use at least 128-bit long keys.

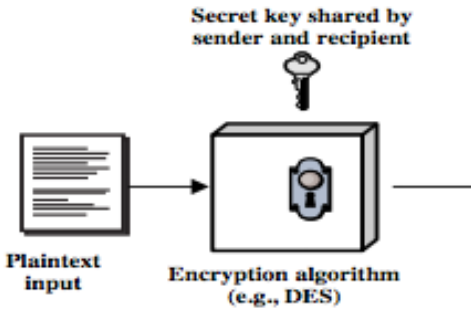

Figure 3.4. Symmetric Key Cryptography [16]

Symmetric key cryptography employs a secret key between two parties. As shown in Figure 3.4 a plaintext input is used as a parameter with the shared secret key in an encryption algorithm. Superficially the encryption and decryption algorithms are black boxes from the parties' point of view. Encrypted data is transmitted through a insecure medium. The receiver of the encrypted message decrypts the cipher text with the shared secret key and calculates the original message.

Modern symmetric cryptographic functions could be categorized under two classes, which are stream ciphers and block ciphers. Stream ciphers encrypt data byte by byte. The most widely used stream cipher is RC4 [15]. Secure Socket Layer (SSL) and Wired Equivalent Privacy (WEP) employs RC4. On the other hand block ciphers encrypt an input data as fixed size blocks, and produces same-sized outputs. The most popular block cipher cryptographic primitive is Data Encryption Standard (DES) [11]. There are also widely used other block cipher algorithms such as Advanced Encryption Standard (AES) [12], RC5 [13] and Blowfish [14].

3.5 Public Key Cryptography

Public Key Cryptosystem (PKC) differs from Symmetric Key Cryptosystem according to key count. PKC uses two separate keys, one of them is the public key the second is the private key. The owner secretly keeps private key, whereas the owner or a trusted third party broadcast the public key. It is computationally infeasible to calculate private key by exploiting the public key.

Figure 3.5. Public Key Encryption [17]

PKC is used for confidentiality purposes, such as encryption and decryption. Also it is used for

authorization purposes such as digital signing and verification. The type of encryption key defines the purpose of the algorithm. If the sender uses the public key for encryption then the algorithm functions for confidentiality purposes as shown in Figure 3.5.

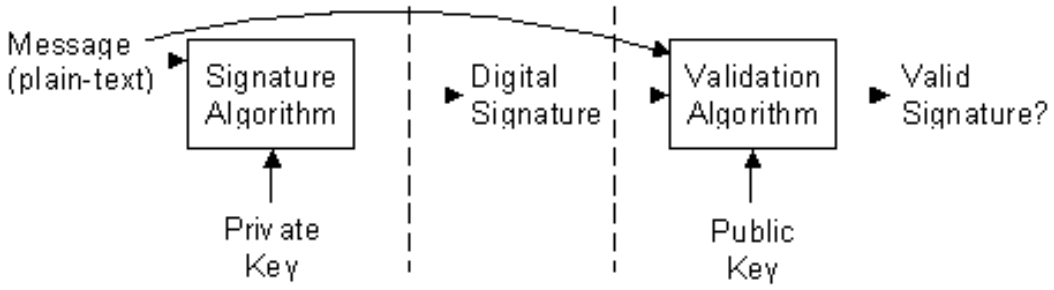

Figure 3.6. Validating a Signature [18]

Authorization and verification purposes are met by using private key as the encryption key as depicted in Figure 3.6. Since no one could know the private key of the owner, only private key owner could produce the encryption of a plaintext encrypted with a private key. This kind of encryptions could be decrypted using the public key. Since the public keys are broadcasted, anyone could verify the digitally signed plaintext. Therefore usage of private keys in encryption does not meet the confidentiality purposes but only authorization purposes.

Digital signature mechanism consists of two parts. The first part is the *Signing* part. The sender processes a plain text with a signature algorithm using the private key. Signature algorithm produces a digital signature. Digital signature does not reveal the plain text unless it is subjected to a validation algorithm that uses the corresponding public key as parameter. The second part is *Verification* part. The receiver processes the digital signature with a validation algorithm by using the public key. Validation algorithm determines if the processed signature is valid.

Some of the widely known, well-regarded asymmetric key cryptographic algorithms are Diffie-Hellman Key Exchange Protocol [19], Digital Signature Algorithm (DSA) [20], ElGamal [21] and the most known one is RSA [22] algorithm.

4. REQUIREMENTS FOR A SECURE AND SEAMLESS MICROPAYMENT SCHEME IN WIRELESS MESH NETWORKS

For a payment scheme designed for Wireless Mesh Networks requires following attributes:

- **Wide Coverage:** Users should be getting service within a large area.
- **Seamless Roaming:** Users should connect and maintain their connection and continue to get service even while they are moving. Designed connection method should apply to different operators. Users should be able to switch between operators as they move without noticing it.
- **Seamless Connection & Roaming:** Users should be able to switch between access points as they move without noticing it.
- **Anonymity:** It should not be feasible to track down a user's network actions from their payments (unless law enforcement requires doing so).
- **Mutual Authentication:** For preventing malicious use of network, both user and network should be mutually authenticated. Moreover, man-in-the-middle and replay attacks must be prevented.
- **Two-way honesty:** clients cannot deny that they did not take service. Operators cannot claim that they provide service more than they actually provided. These are to be guaranteed using strong cryptographic protocols.
- **Preventing Double Spending:** A payment token should not be able to be used to get more services that its value. In particular, the payment token should not be used twice or more.
- **Untraceability:** It must not be possible to relate connection sessions of the users with other connection sessions. In this way, higher level of privacy could be provided.
- **Performance:** System should work fast and effective.

4.1 Requirements of the Network

Secure and seamless pre-payment system for Wireless Mesh Networks will not only consists of mesh backbone but also Wi-Fi clients and wired servers. Mesh backbone will basically relay the packages from clients to server to make the users able to get service.

Servers of the operators are wired and will be communicated via regular 802.3 Ethernet protocol in its local area. Mesh backbone will communicate within itself using IEEE 802.11s protocol. Clients will use IEEE 802.11a/b/g Wi-Fi protocols to connect to the access points/mesh routers.

4.2 General Overview of the Proposed Scheme

The proposed system supports user identification, authentication as well as authorization and accounting. The main objective is to design and develop a secure payment infrastructure for WMNs that also considers users' privacy and fairness. The basics of the system model, roles, entities and requirements have been identified in Deliverable 1. As mentioned there, our system model assumes mobile clients and operators, who will be charging the service they give. The operator's mesh backbone is made of several mesh routers, which are actually Access Points (APs) with IEEE 802.11s support. This backbone is connected to operator's server via a gateway. There exists a TTP (Trusted Third Party), which may be reachable through operator. These system components are listed; together their icons used in the protocol figures, in Table 4.1.

Table 4.1. System Entities

	Mobile user (client)
	Access Point (AP) with mesh routing capability. From now on in this document, it is called as AP, but please note that it also has routing capability.
	Mesh backbone of the operator
	Gateway (GW) that connects the mesh backbone to outer world and also to the operator's server
	Operator's server (OP). Keeps necessary logs and user info.

	Trusted Third Party (TTP). Payment related logs are mostly to be generated by the TTP.

Since the clients are mobile, they may hand over among different mesh routers (i.e. access points) of the same operators. They may also roam among different operators, not only due to coverage reasons, but also for having a better quality service. Our system aims to have seamless mobility and seamless roaming for payment purposes such that when the client gets service through a new AP or switch to another operator, authentication and authorization are not performed from scratch.

From security point of view, we aim to have mutual authentication between client and the network in our protocols. Anonymity of the clients and untraceability across different usage periods (a.k.a. unlinkability) are privacy related goals of the protocols.

From payment point of view, our main aim is to have a fair system in which all the claimed transactions bear cryptographic proofs. In this way, the clients cannot repudiate using a service and the operators cannot claim for services that they do not provide. The latter is especially important during inter-operator settlement; it is also important to resolve client disputes.

The protocols detailed in this deliverable are designed by considering the abovementioned requirements. The symbols used in this document are given in Table 4.2.

Table 4.2. The List of the Symbols

$\oplus$	XOR operation
$\|\|$	Concatenation
$E_K(X)$	Encryption of Xusing the key K
$D_K(X)$	Decryption of X using the key K
$h^n(X)$	Taking hash of Xn times

$HMAC_K(X)$	Taking HMAC of X using the key K
H_i	i^{th} element of the hash chain (usage order)
$PU\text{-}TTP$	Public key of TTP
$PR\text{-}TTP$	Private key of TTP
AP_i	i^{th} Access Point or its identity
OP_i	i^{th} Operator or its identity
$PU\text{-}AP_i$	Public key of AP_i
$PR\text{-}AP_i$	Private key of AP_i
SN	Serial Number
N_X	Nonce created by entity X
PA	Previous Alias
NA	New Alias
$cert_i$	Public key certificate of AP_i
IV	Initialization Vector
TS	Timestamp
CR	Connection Request
DR	Disconnection Request
RR	Roaming Request
CAR	Change Alias Request
$MobReq$	Mobility Request
RP	Response (used in various protocol as positive acknowledgment)
DA	Disconnection Acknowledgement
$RAck$	Roaming Acknowledgement
$MobResp$	Mobility Response

4.3 Network Topology and General System Design

Secure and Seamless Pre-Payment System employs previously explained system entities. The system entities are assumed to be located in a metropolitan area. While access points establish a mesh backbone and wait for clients to connect to them, gateways transmit the packets received from the access points to servers of the operators.

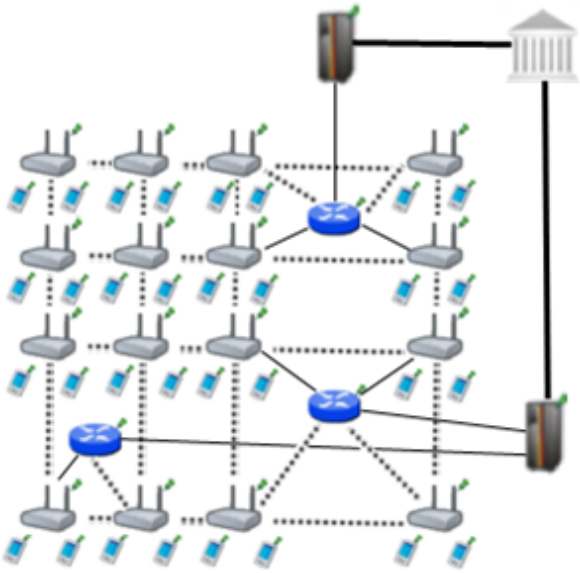

Figure 4.1. Network Topology

Figure 4.1 shows the topology of the network and connections between entities. Connection between serving access points is wireless and they use IEEE 802.11b/g Wi-Fi protocol, and they use IEEE 802.11s protocol [25]. The mesh backbone emulates a cloud from the mobile user's perspective. It is a black box; which receives packets from mobile user and delivers them to the gateway in a multi-hop manner. Mesh backbone uses Hybrid Wireless Mesh Protocol (HWMP) [26], which is a hybrid routing protocol, which has routing tables.

Connection medium between mesh backbone and gateway (GW) is either wireless or wired. GWs and operators communicate through wired connection. The connection between an operator and TTP is also wired. These connections use 802.3(Ethernet protocol) [27].

4.4 Connection Card Structure

Connection Card is the main deed that clients buy from operators and use to get

Internet service. We use a prepaid system, in which connection cards include credits as tokens. Hash tokens are generated using hash chains as discussed below. Connection cards also have unique *Serial Numbers* (*SN*), which are to be used for alias computation.

Tokens for getting Internet service are basically links in a hash chain. For each set of tokens, the operator picks on a random *Initialization Vector* (IV) and takes hashes of it many times. The number of hash operations is actually the number of token in a set. For example, if the client wants a hundred hash tokens, then the hash of IV is taken hundred times. More formally a hash chain with, say 100 tokens are constructed in the following way.

$$H_0 = h(H_1) = h^{99}(IV)$$

$$H_1 = h(H_2) = h^{98}(IV)$$

$$H_2 = h(H_3) = h^{97}(IV)$$

...

$$H_{98} = h(H_{99}) = h^2(IV)$$

$$H_{99} = h(IV)$$

H_0 is the first token to use. Then we use the token in the increasing order of token index. In this way, we exploit one-way property of hash algorithms such that an attacker cannot learn the next token even if she knows the previous ones.

The operators inform the TTP (Trusted Third Party) about the associations between SN and corresponding H_0 so that TTP validates them as needed in the protocols.

Connection Cards are refillable with hash tokens, which are to be sold by the operators. We assume a free market strategy in the marketing of the hash tokens. The prices or campaigns related for the marketing of hash tokens are to be decided by the operators. In other words, operators would compete with each other to sell hash tokens. They also compete with each other to provide high-quality service for broadband access in the WMN since the users are assumed to have free roaming.

Serial Number SN is a 16 digit alphanumeric and case sensitive value. With this setting, the system is able to support up to 62^{16} users. Hash tokens are to be generated

using SHA-256 hash algorithm; hence they are 32 bytes long.

Considering current technology, smart cards are suitable tools to be connection cards. A simple Connection Card with 4 KB memory could store a SN and more than 100 hash tokens.

4.5 Alias Computation

Aliases are temporary identifiers for clients. They change frequently using a secure protocol. Anonymity is achieved by changing aliases as previously stated way however it is durable to some extent.

The serial number (SN) of the CC, which is bought from an operator, will be used as a base for client's aliases. An alias will be computed by performing the following operations:

1. Client will pick a random 128-bit unsigned number and call it his nonce N_{CL}.
2. Perform XOR operation with SN and his nonce; take the hash of the output. $h\,(SN \oplus Nonce) = Alias$
3. Client will use this alias whenever his identity is required.

One may argue that this kind of alias computation would run a risk of producing same alias for several users. However making TTP to check the proposed alias to be a unique one solves this problem. This check is done in Change Alias protocol, which will be mentioned in Section 6.

The nonce values used in substitution of the aliases are to be sent in encrypted messages to the TTP in the related protocol. Therefore only the client and the TTP can relate the aliases originated from a particular SN.

One may argue that this kind of alias computation would have a risk of causing same alias for several users. Aliases are 128-bit values; even if it is a very small possibility to have the same alias with another client at a given point of time, there is still a nonzero probability. The problem is addressed by making TTP to check proposed alias to be a unique alias at that point of time. This check is embedded in related protocol, which will be described later.

5. EVOLUTION OF SSPAYWMN

The idea of SSPayWMN is firstly found and rooted in [38}. The motivation behind this thesis is same with [38], to build a secure and seamless micropayment system for wireless mesh networks. We have taken over a robust and consistent system and improved it.

The topology of the network we use is analogous to previous network topology. System flow is very similar except some changes we have inserted. The packet flow that used was from access points to operators, with mesh backbone and gateways in-between. The connection mediums between clients to the gateways were not encrypted and insecure lines. Gateways and operators however were connected with secure links. We did not change the assumption of secure lines between gateways and operators but also brought symmetric cryptography security between gateways and access points. New system achieved a securer mechanism.

Simulation environment of current version of SSPayWMN uses the same tools. Mesh routers, gateways and servers are the same with the ones that were used in previous versions. A change on devices did not take place in this thesis because it was necessary to compare the results by keeping the control group. Previous version used RSA-1024 however we have used RSA-2048 to improve the security of the system. We have inserted symmetric key encryption using AES-128.

Most of the previous protocols are changed and there are some new protocols also. However some protocols did not need any improvement and they were still necessary. Access Point Authentication protocol is the only example of this situation. Access Point Authentication protocol is not exposed to any modification.

The main addition to the system is a Trusted Third Party (TTP). To bring an ultimate authority to the system was necessary for the requirements, which will be mentioned later. The usage of TTP and it's servers provides credibility. In the system operators settle in the system by communicating with TTP. Firstly clients pre-pay to TTP for the Internet service they are going to receive. Operators receive their payment from the TTP as they show logs of their service. Previous version of SSPayWMN used client to operator and vice-versa packet delivery in end-to-end protocols, we did not change the order but inserted TTP at the end of the topology, therefore end-to-end packet deliveries in the new version of SSPayWMN has

TTP at the end of end-to-end protocols.

In [38] some system entities were assumed to have public keys but an algorithm to distribute these keys was lacking. Existence of a TTP brings a possibility to use certificates in the system. SSPayWMN employs certificates for distribution of public keys. The distributed public keys could be broadcasted since they are signed by the TTP.

The settlement of the operators has been improved. Previous version of SSPayWMN did not clarify how the access points of the operators will be placed in the metropolitan area and the distribution of access points between operators. In SSPayWMN simulations two operators were assumed to exist. Operators share the area in-between and compete to serve the users with stronger access points. There is a rivalry between operators since clients connect to stronger access points. An operator with low amount of investment for the system could not survive since high amount of access points are needed to serve in a wide range. The clients would not connect to an access point with a low signal rate if there is another access point with high signal rate in their range. As it will be explained later, there is no difference between operators in the sense of payment for the clients.

The main deficiency in [38] was anonymity and untraceability of the clients. Previous version of SSPayWMN revealed user identities and user actions to any adversary. These properties of the system should be provided by the system as long as there is no request for user actions and identity by a formal authority such as a state. In case of a formal request from the TTP, it should provide all the logs to the formal authority. Client identity or actions should be provided in a readable plain text format. Current version of SSPayWMN provides anonymity and untraceability. Using aliases as client identity provides anonymity in the system. The aliases are changed periodically therefore only the actions between the periods are open to traceability. The time period and detailed implementation will be explained in later chapters of the thesis.

In the previous version roaming between operators was costly therefore it was not seamless to the client. Clients were customers of a particular operator, which suggests an overcharge in the case of roaming to another operator. Current version of SSPayWMN makes every user of the system a client to the TTP. The TTP pays operators for their services. The new setting enables SSPayWMN to provide seamless roaming. The proposed system does not suggest an overcharge for roaming clients. Nonetheless there are still some small changes

between the protocols, which will be mentioned in the following chapters of this thesis.

Internet service providing is simplified in current version. The pre-payment was for a total packet size. In contemporary version clients buy hash tokens that ensure Internet service for a predefined period of time. Seamless micropayment is preserved. Furthermore we have increased the Update Packets interval in the system because the changed way of Internet service providing for pre-defined time periods suggested that we do not have a small Update Packets interval. In the new setting it is enough to have a time interval, which is slightly bigger than the multiplication of time value for one hash token by two. In a situation of a client that does not send the next hash token for a time would mean that client is dropped from the system. Therefore the heavyweight on the system caused by update packets is reduced significantly and with the new settings it is easier to handle the dropped clients.

In the earlier version, simulations of the proposed scheme were done considering the same type of user and multiplying the same behavior to construct a result. In this thesis we have divided clients into groups considering their place in the society. We keep the randomness in the system by using random number generators but the randomness of the system is affected by the client properties. Speed, traveling distances, client's frequency of system usage are all affected by the client types. Earlier version of SSPayWMN was lacking simulations of real-life situations, which are covered in the current version. Possible situations in real-life such as the diversity in mobility or system usage in time were not covered in [38]. There were simulations of empty hours and a burst scenario but these simulations were superficial. Burst scenario only covered users trying to authenticate themselves into the system on the same time. A rush hour scenario was lacking in which clients try to authenticate themselves at the same hour but necessarily on the same exact moment. Current version covered most of the possible scenarios within a day except a scenario of a natural disaster or a hazard. Real-life scenarios of the current version cover an ordinary day.

In the previous version of SSPayWMN clients' mobility patterns were random. The movement actions of the clients were not dependent on a logical reason. The lacking of a systematic explanation on client mobility caused the distinction between the simulations and possible real-life performance of the system. There was basically not enough evidence for the system to work properly in a case of client mobility in an hour where every client in the

network is trying to return to their homes after 06.00 pm. Previous version employed the Random Way Walk Model [39] of the network simulator. In the new version of SSPayWMN clients move from one point to another for a purpose. The new setting brings more realistic results from the network simulations. The new network simulations are designed considering a real metropolitan area. We have used a similar mobility modal with Manhattan Mobility Model [41]. The setting of the roads is a grid, and the clients move from one location to another by using these roads. We have replaced the movement probabilities of the original Manhattan Mobility Model with the probabilistic values of the client's movement probability values in SSPayWMN. Additionally the recent simulations cover a larger area with more access points than the previous simulations. A larger area to simulate enables to cover more situations than a narrow area simulation. Previous simulations had 32 access points with 16 gateways. Recent simulations have 100 access points with 32 gateways.

A very significant improvement on [38] is the change of the network simulator used to simulate the system. The previous project was simulated using the Discrete Event Simulator: OMNET++ [40]. OMNET++ has offered GUI support and strong network simulation tools but it was lacking IEEE 802.11s [25] support. Therefore, an ad hoc network simulation was executed to mimic the simulation of a real wireless mesh network. Converting the previous simulation results to a new network simulator was not feasible because there was no connection between OMNET++ and another network simulator. It was inevitable to implement the system from scratch on a network simulator, which has IEEE 802.11s wireless mesh network support. We have chosen network simulator 3 (ns-3) [6]. Ns-3 supports IEEE 802.11s protocol moreover it has helpful examples for mesh networking. However ns-3 did not support inter-networking for mesh networks. It was infeasible to implement inter-networking and bridging functionalities for mesh networks on ns-3 since there was not enough manpower. We have implemented virtual bridges between network nodes, and write every packet sent or received on text files. Every node in the system checks for packets to send in the text files. We have implemented the system in multiple dimensions; every node in the mesh backbone had two interfaces in our design. We have neglected the delay of passing the packets from one interface to another that's how we have mimicked inter-networking and bridging functionalities of wireless mesh networks.

Detailed and more realistic simulations brought more detailed results with quality. In the previous version of SSPayWMN there were no results for specific protocols and because

of the fact that there were no client roles, there was not any result for a specific client type. It was not possible to see protocol performances. The strong or weak sides of the system were not revealed. In the past simulation results there were 4 performance metrics. They were End-to-End Authentication Latency, Server Service Time, Connected Node Count and Connected Node Count per AP. These performance metrics were not enough to evaluate the system because they are not protocol specific and they do not offer any information about client types that could be used for marketing purposes by the operators. As they will be explained at the later chapters of the thesis, up to date version of SSPayWMN offers very detailed simulation results. Every protocol has a unit simulation result and a real-life scenario simulation result. These results are presented on charts and they show the average delay for the protocols to run and it could be analyzed considering a 24-hour usage of the system since both unit and real-life scenario simulations are run for virtual 24 hours. Furthermore there are specific results for client types. It is distinctively stated how many hours of service did (e.g.) students received in a day. It also possible to determine the length of connected time of a client type on a specific part of the day by looking at the probability values of the client types. Present version of SSPayWMN covers this significant deficiency of the old version.

Considerable improvements are committed on the system and presently it is more powerful than it was. Conclusively, the system supports more features and it has more realistic simulations.

6. PROTOCOLS OF THE SYSTEM

6.1 Initial Authorization and Reuse of a Connection Card

Initial Authorization is the beginning for system usage. Whenever a client purchases new hash tokens from the TTP, she will need to authorize herself to TTP. Initial Authorization Protocol, shown in Figure 6.1, achieves mutual authentication and authorization of the user.

The clients may disconnect before using up all the credits in a connection card. *Reuse of a Connection Card* (Reuse-CC) protocol allows the clients to connect using the remaining credits in a card. Reuse C.C. protocol does not differ extensively from *Initial Authorization* protocol. The main difference is instead of sending first hash token; the client sends whichever token is the next one. Alias will change before the protocol starts. Both protocols compute new aliases before sending the Connection Requests (CR). The crucial point here is that TTP should be able to update last hash value entry of the client in the database and associate it with the new alias.

In Figure 6.1, connection between client and serving access point (AP_S) is Wi-Fi (IEEE 802.11b/g). The access point is a member of a mesh backbone and a particular access point is to be selected according to its transmission power. Since it is assumed that all access points have the same attributes, the serving access point is the closest access point to the client.

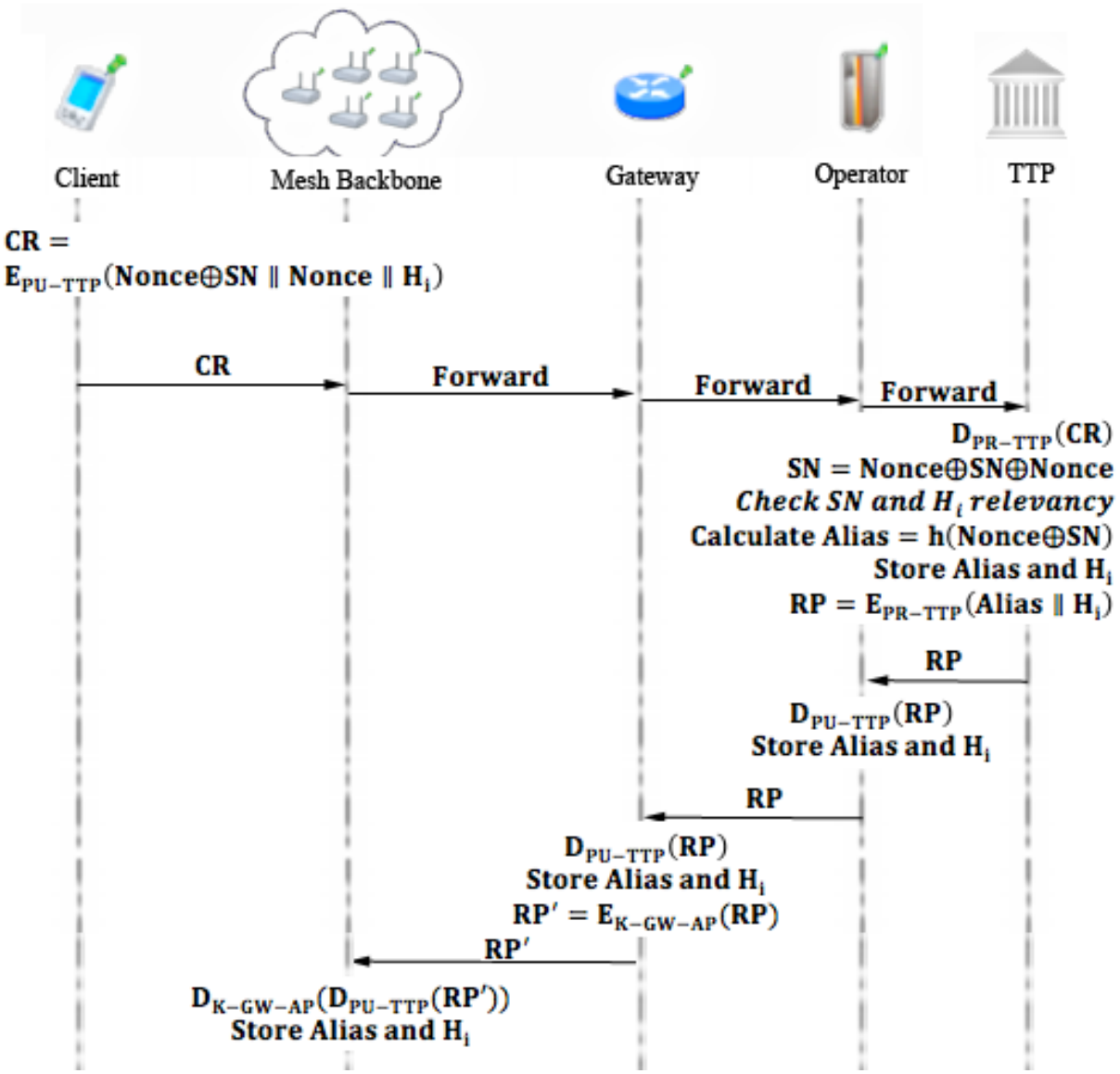

Figure 6.1. Initial Authorization and Reuse of a Connection Card

Mobile clients introduce themselves to the operator using *Initial Authorization* protocol. $H_i = H_0$ in Initial Authorization protocol, $H_i = H_k \; \forall \; k > 0$ in Reuse of a Connection Card protocol. TTP already knows mobile user's serial number (SN) and the first element, H_0, of her hash chain. The mobile user does not want to reveal her SN to any adversary because that SN will be used continually; it is as valuable as mobile client's identity. To achieve anonymity, the mobile client computes an alias and uses this value instead of SN. The mobile client will change her alias periodically as she continues to get service (Change Alias protocol will be explained later).

Initial Authorization and Reuse-CC steps are described below.

1. Client computes an alias using a nonce N_{CL} that she generated.
    a. $Alias = N_{CL} \oplus SN$
    b. $H_i = h^{T-i}(IV)$ (The CC is assumed to have T credits)
    c. $CR = E_{PU-TTP}\ (N_{CL} \oplus SN \parallel N_{CL} \parallel H_i)$
    d. Client sends this CR to AP_S.
2. AP_S receives the connection request and relays the request through mesh backbone.
3. Gateway receives the CR and relays it to the operator.
4. Operator relays CR to TTP.
5. TTP receives the connection request (CR) and decrypts it using its private key.
    a. $D_{PR-TTP}\ (N_{CL} \oplus SN \parallel N_{CL} \parallel H_i) = N_{CL} \oplus SN \parallel N_{CL} \parallel H_i$
    b. TTP checks alias' uniqueness within its database of users, it would make the client start over the protocol if alias is not unique.
    c. It computes $N_{CL} \oplus SN \oplus N_{CL} = SN$.
    d. TTP checks SN and H_0 association. Store $N_{CL} \oplus SN$ and H_i
    e. TTP computes $RP = E_{PR-TTP}\ (N_{CL} \oplus SN \parallel H_i)$
    f. TTP sends RP to the Operator.
6. Operator receives RP and verifies the signature using public key of TTP.
    a. The Operator gets $N_{CL} \oplus SN$ and H_i and stores these values. The value of $N_{CL} \oplus SN$ is the client's alias until she changes it.
    b. Operator sends RP to the gateway.
7. GW receives RP and verifies the signature using public key of TTP.
    a. GW stores $N_{CL} \oplus SN$ and H_i.
    b. GW uses the shared secret key with AP_S and calculates $RP' = E_{K-GW-AP}(RP)$
    c. GW sends RP' to AP_S through mesh backbone.
8. AP_S receives RP' and decrypts it using the shared secret key with GW.
    a. AP_S verifies the signature using public key of TTP.
    b. It calculates $N_{CL} \oplus SN$ and H_i and stores these values.

The wired links are secured however the communication between GW and APs are insecure; therefore the packets that are sent through this medium are encrypted with shared secret keys between GWs and APs.

6.2 Access Point Authentication

After authentication processes of the client with the TTP, a second authentication step begins. Client and access point will mutually authenticate each other for safe communication; this protocol ensures the feature -Mutual Authentication- of SSPayWMN.

Figure 6.2 describes the protocol briefly.

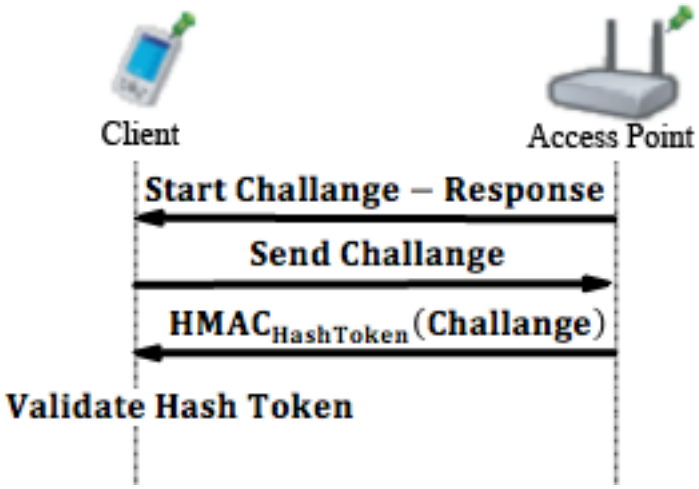

Figure 6.2. Access Point Authentication

1. AP_S sends a challenge request to the client, which started connection.
2. When client receives this challenge request:
   (a) Client drops the packet if it is not the AP_S that she sent connection request.
   (b) Client drops the packet if there was not any CR.

   If (a) and (b) are 3 invalid then the client sends a 128-bit challenge to the AP_S.
3. AP_S takes the HMAC of this challenge, and uses relevant hash value (here $HashToken$, but it could be any H_i if the authentication protocol runs after the Initial Authorization or Reuse-CC protocols) as the key of HMAC.
   (a) $Response = HMAC_{HashToken}(Challenge)$
   (b) AP_S sends $Response$ to the client.
4. Client also takes the HMAC of the challenge and uses the stored hash value ($HashToken$) as the key. Then it compares the result with the one that access point sent.

If it is authenticated, client starts to use access point to get Internet service.

6.3 Packet Transfer

After mutual authentication of client and AP_S client starts to send packets as shown in Figure 6.3.

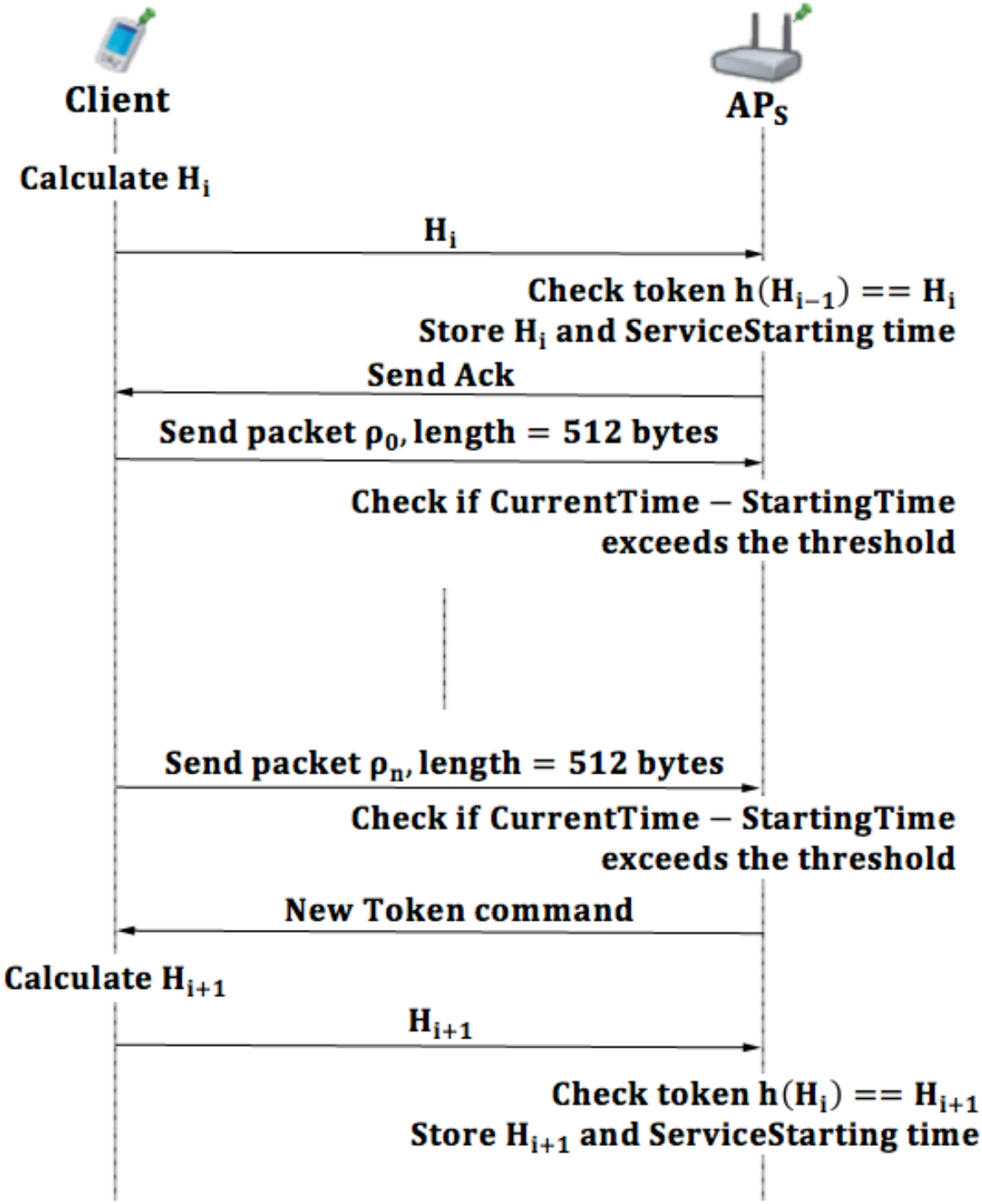

Figure 6.3. Packet Transfer

1. Client starts the session with the first hash token (in this case current has value is H_i)

of the remaining hash chain.

2. AP_S receives H_i, and updates client's service starting time.
   (a) Checks if $h(H_{i-1}) == H_i$
   (b) If true sends acknowledgement (Ack) to client and updates currently used hash value as H_i.
3. Client sends first 512-byte data packet p_0.
4. If the client gets served for over the threshold value (5 minute interval is used in simulations) then the AP asks for the next hash token.
5. The steps between 1-4 are repeated as long as client gets Internet service.

6.4 Changing Alias

Anonymity property is easily achieved by using aliases, but complicated part is achieving untraceability. The aliases should change on a basis that an adversary, who knows a certain client's alias, could not be able to trace client's activity on her home network, and also could not trace her movements among the operators or access points.

To be able to change alias in a safe way, client needs to communicate with TTP but interrupting TTP very often would slow down the entire operation due to extra delays caused. Therefore periodic changes of aliases are mandatory and these changes are achieved by making access points to ask all of the active clients for new aliases after a certain period of time. Attackers or access points themselves would know that aliases are changed but would not know the mapping between old aliases and the new ones. Such a protocol is also used in Mix Networks [28].

Simultaneous alias changes aim to prevent attacks that would aim to analyze network traffic of access points and examine connection requests. Enforcing alias change by the access points, a more generalized control over the clients is achieved. Attackers could not understand which client wanted to change her alias, because all the clients getting service from a particular access point have requested to change their aliases at that particular time.

The client should request changing alias, because client and the TTP should be the only parties who know association between an alias and a client's SN.

Alias Change Timer is a local timer that runs on every Access Point. All of the timers

are set roughly to the same time manually. System designer decides on the time value on which the access point will count down from (50 minutes of time period is used in simulations). The timer period is updateable by the TTP. TTP knows every access points' public key, it could send new interval by encrypting the new value with the public keys of the access points. However this process is not covered in simulations.

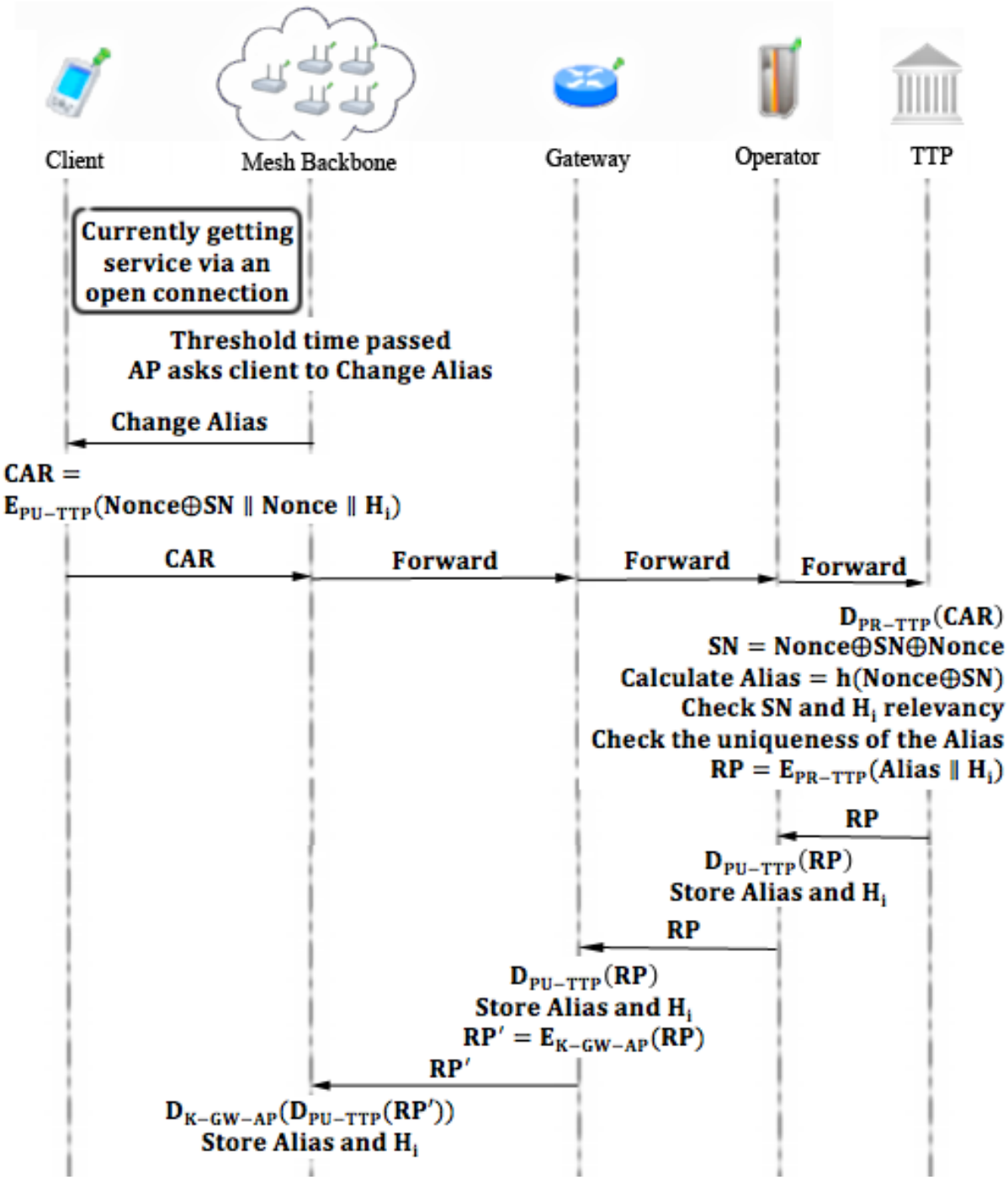

Figure 6.4. Changing Alias

Changing Alias Protocol is shown in Figure 6.4 and described below.

1. Client continues to get service, in other words uses the *Packet Transfer* protocol.

   When the Alias Change Timer countdown finishes, Access Points broadcast "Change Alias" command to all of their clients. The interval value is a system parameter; 50 minutes of interval value is used in the simulations.
2. Client receives "Change Alias" command.
   a. Client computes a new alias by picking up a new random nonce N'_{CL} and computing $N'_{CL} \oplus SN$.
   b. Client forms a Change Alias Request (CAR)
   c. $CAR = E_{PU-TTP}(N'_{CL} \oplus SN \parallel N'_{CL} \parallel H_i)$
   d. The client sends the CAR to AP_S.
3. AP_S receives CAR and relays it to the GW via mesh backbone.
4. Gateway forwards CAR to operator.
5. Operator forwards CAR to TTP.
6. TTP receives Change Alias Request (CAR) and decrypts it using its private key.
   a. $D_{PR-TTP}(N'_{CL} \oplus SN \parallel N'_{CL} \parallel H_i) = N'_{CL} \oplus SN \parallel N'_{CL} \parallel H_i$
   b. TTP checks for new alias' $h(N'_{CL} \oplus SN)$ uniqueness and starts over the protocol if not unique.
   c. TTP computes $N'_{CL} \oplus SN \oplus N'_{CL} = SN$.
   d. It checks SN and H_i association and stores $Alias = h(N'_{CL} \oplus SN)$ and H_i.
   e. It computes $RP = E_{PR-TTP}(Alias \parallel H_i)$.
   f. TTP sends RP to operator.
7. Operator receives RP and verifies the signature using public key of TTP.
   a. The operator receives $Alias$ and H_i and stores these values.
   b. Operator sends RP to the GW.
8. GW receives RP and verifies the signature using public key of TTP.
   a. The GW receives $Alias$ and H_i, and stores these values.
   b. The GW encrypts the RP and calculates $RP' = E_{K-GW-AP}(RP)$
   c. GW sends RP' to the AP_S.
9. AP_S receives RP' and decrypts it as follows:
   a. $D_{K-GW-AP}(RP')$
   b. The AP_S verifies the signature using public key of TTP.

c. The AP_S reveals $Alias$ and H_i and stores these values.

6.5 Update Packets

In standard flow of the system, after authentication, access points handle the accounting. Because of the fact that access points keep the last alias and token of the client they are able to validate next token by performing hash operation to the token they kept and compare it with new coming hash token. However it is essential to send periodic updates to the TTP to provide stability in the system in the case of client drops.

Access points keep track of ongoing communications, after some time passed without update from a user it send disconnection request by itself. When access points broadcast change alias commands they delete all the record related to previous connections therefore they do not send unnecessary disconnection packets to TTP.

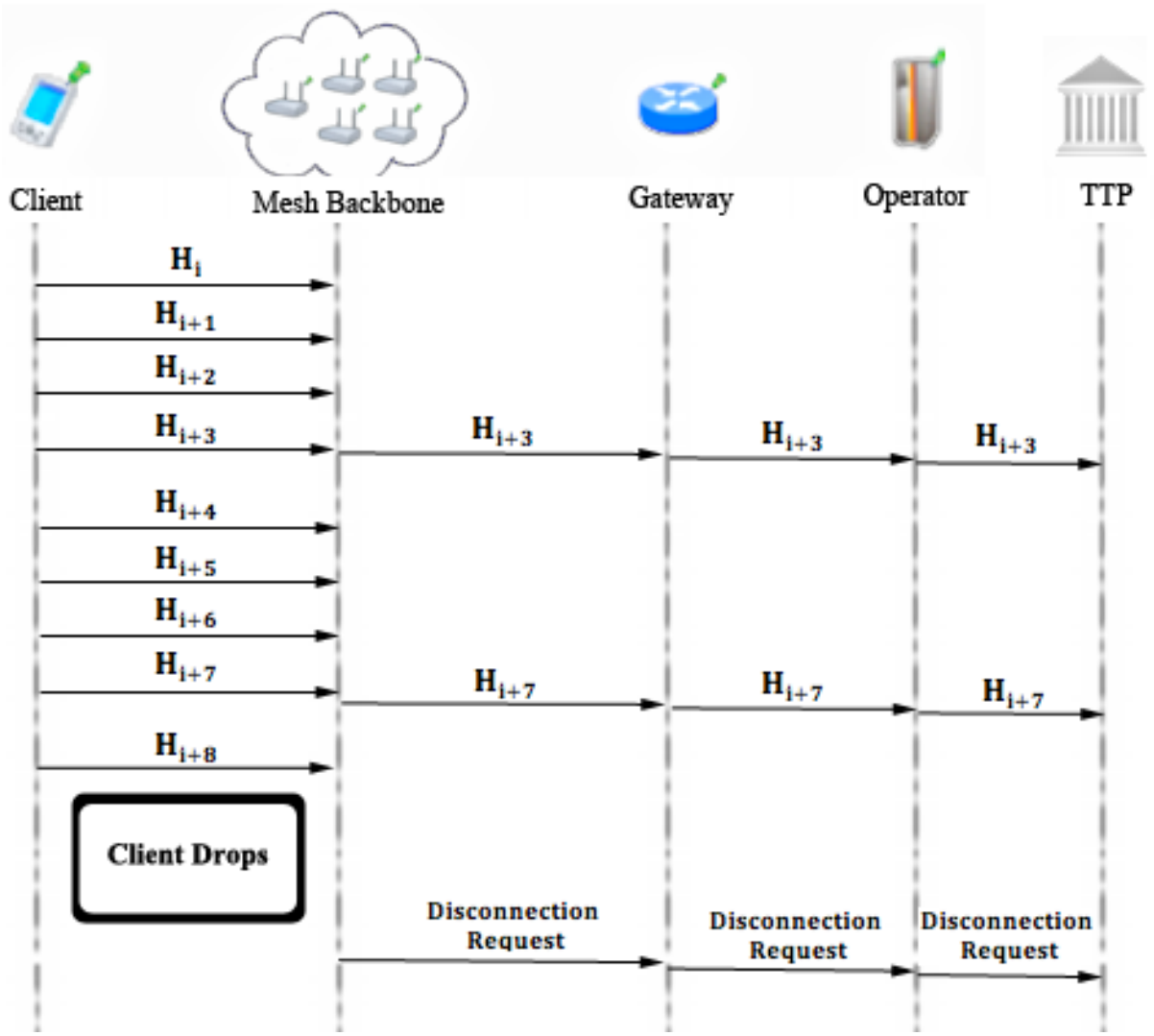

Figure 6.5. Update Packets

Protocol design of Update Packets protocol is shown in Figure 6.5 and the details of the protocol are explained below.

1. After client sends the first token, the access point starts to count the time passed. After t units of time (value of t is a system parameter, 11 minutes of an time interval is used in simulations), access point encrypts the Alias and lastly used hash token using the public key of the TTP and sends this cipher text to the GW.
2. The GW receives the update packet and forwards it to TTP through related operator.
3. TTP receives the update packet and decrypts the packet using its private key. TTP updates the last token used by the client.
4. In a case of client drops from the network, access point concatenates the Alias, hash value and a time stamp and encrypts them with the public key of TTP.

Sends it to TTP as a disconnection request from the client.

6.6 Disconnection

To be able to run Reuse-CC, the client has to run a proper disconnection protocol. The Update Packets protocol brings stability to the system in case of a connection interruption, but the main assumption is that most of the users will be disconnecting from the operator using the disconnection protocol that we explain in this section and in Figure 6.6.

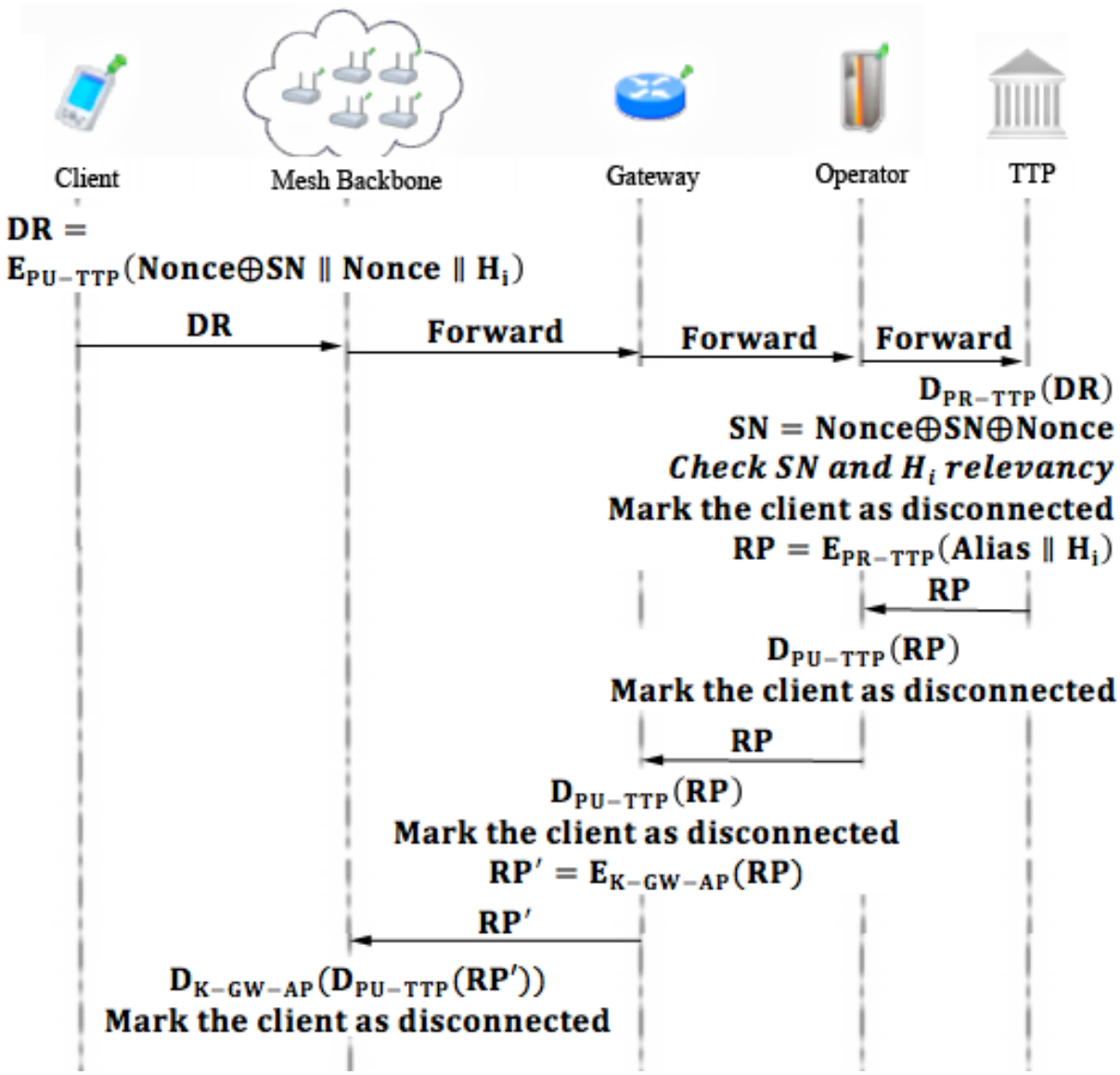

Figure 6.6. Disconnection

Disconnection protocol is described below.

1. Client forms a disconnection request
   $DR = E_{PU-TTP}\ (Alias \mid\mid H_i)$
   Client sends the packet to the AP_S.
2. AP_S relays DR to the mesh backbone, to make it reach to the GW.
3. GW receives and forwards the DR to the Operator.
4. Operator receives and forwards the DR to the TTP.
5. TTP receives the $Alias$ and H_i. It checks the association between the $Alias$ and the hash token; if the association holds, then it computes a disconnection acknowledgement (DA).
   $DA = E_{PR-TTP}\ (Alias \mid\mid H_i)$
   TTP sends the DA to the Operator.
6. Operator receives DA, verifies the signature on it and marks client as disconnected.
   Operator relays DA to GW.
7. GW receives DA, verifies the signature on it and marks client as disconnected.
   It relays DA to the mesh backbone.
8. AP_S eventually gets the DA, verifies the signature on it and disconnects the particular client, which corresponds to the $Alias$ it received. Ideally access points are assumed to delete all information about the past connections for the sake of anonymity and untraceability. However if operators decide to trace user's actions then they could do so for a limited time until the client changes it's $Alias$.

6.7 Distributing Access Point Public Keys

Achieving seamless mobility in home operator and also to support seamless roaming, a public key distribution mechanism is integrated in SSPayWMN system.

In Figure 6.7, a generic model for public key distribution is shown. This protocol has two parts; one is certificate generation for access point public keys, the other one is distribution of the public keys. The part between operator and the TTP is offline. This part of the protocol runs during set-up, before the deployment of the access points in the field.

If an operator wants to add a new access points to the metropolitan area then it should perform the same protocol but his time only for the new access points.

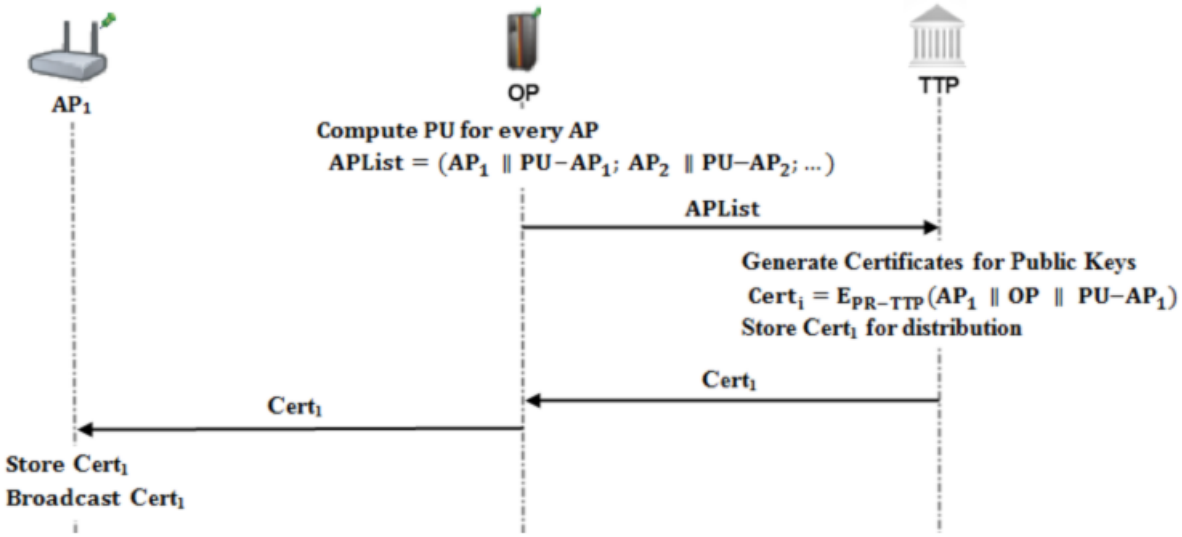

Figure 6.7. Distributing Access Point Public Keys

Distributing Access Point Public Keys algorithm is described below.

1. Operator generates public/private key pairs for the access points in its mesh backbone and embeds these keys to them before the deployment.
    - Operator forms an access point list ($APList$); which consists of access points and their corresponding public keys.
    - Operator sends this list to the TTP through a secure channel or in offline manner.
2. TTP receives the $APList$ and starts to generate certificates for every access point and public key pair.
    - Certificates are formed as:
    - $Cert_i = E_{PR\text{-}TTP}(AP_i \parallel OP \parallel PU\text{-}AP_i)$
    - TTP stores these certificates for distribution.
    - Other protocols are employed (such as *Initial Authorization* or *Reuse-CC* protocols) of SSPayWMN for certificate distribution. Suppose an AP does not possess its certificate. In such a case whenever this access point gets a connection request it will concatenate a certificate request to the packet. When the TTP receives such a request, it concatenates corresponding certificate to the connection response. Then, TTP sends the connection response and $Cert_i$ together to the operator.
3. Operator receives the connection response and the certificate and relays these packets

to the access point through gateway and mesh backbone.

4. Access point receives and stores its certificate and broadcasts it to the nearby access points.

6.8 Seamless Mobility and Roaming (Payment Related)

Seamless Mobility and *Roaming* protocols are run whenever the client changes the serving access point. The running protocol is called *Seamless Mobility* if the new access point belongs to the same operator as the previous access point. If the operators differ, then the protocol is called *Seamless Roaming.*

Every access point has its public/private key pair and ability to broadcast it's public key, seamless mobility in current operator and roaming could be handled in a seamless way without running the authorization process from scratch. As it is shown in Figure 6.8, client gets a signed handover ticket from its old access point and uses this signed ticket to maintain to get Internet service from a new access point.

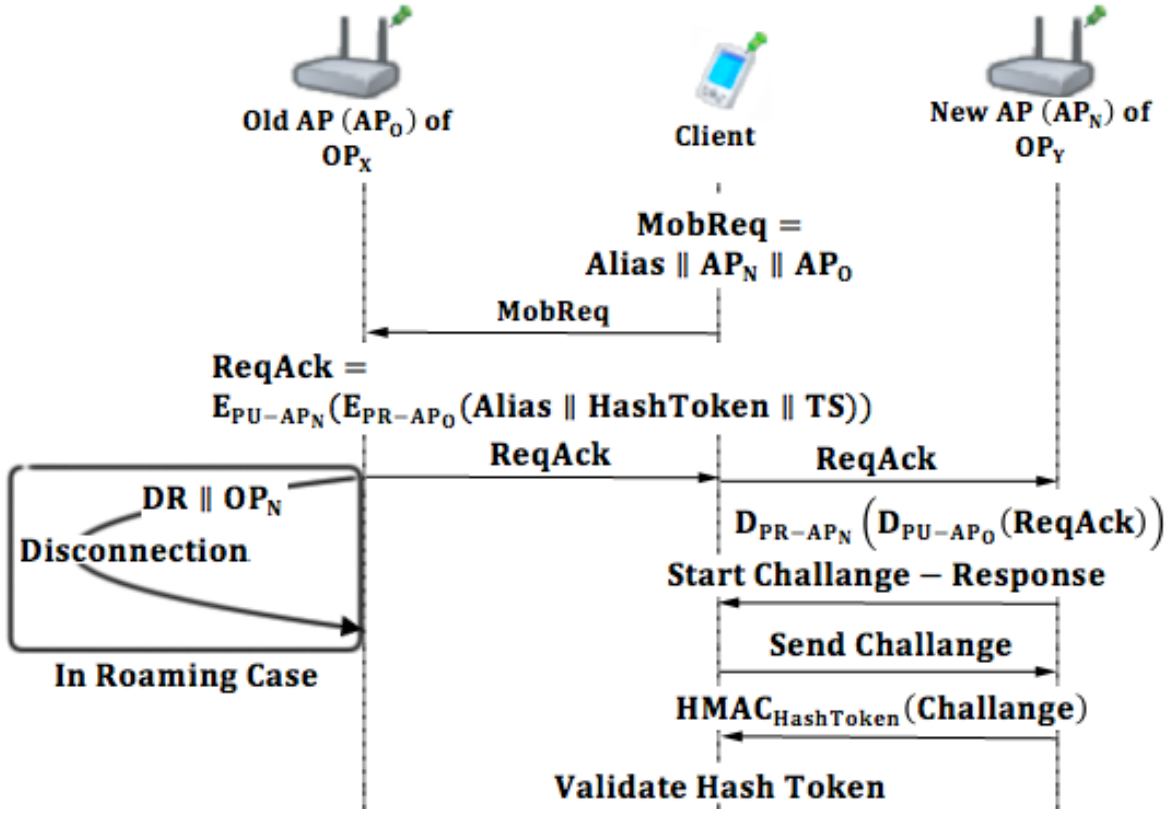

Figure 6.8. Seamless Mobility (if $X = Y$) or Roaming (if $X \neq Y$)

Seamless Mobility and Roaming protocol is shown in Figure 6.8 and described below. In this protocol, the client would like to switch from its old operator (OP_O) to a new one (OP_N). In this setting, AP_O is the last access point that the client got services from OP_O. AP_N, is the access point that the client would like to continue to get services in OP_N network.

1. Client sends a Mobility Request ($MobReq$) to AP_O.
    - $MobReq = Alias \parallel AP_N \parallel OP_N$
2. AP_O receives RR and forms a Roaming Acknowledgement ($ReqAck$).
    - $RAck = E_{PU-AP_N}(E_{PR-AP_O}(Alias \parallel HashToken \parallel TS))$
    - AP_O sends $ReqAck$ to the client.
    - $ReqAck$ consists of the mobility ticket that the client uses to get services from the AP_N. It is signed by AP_O and encrypted for AP_N.
3. If the new operator is different than the previous operator, then AP_O starts the disconnection protocol for the client after sending $ReqAck$.
    - This disconnection protocol runs in parallel with the roaming protocol. Thus it does not put an extra delay in roaming. Old operator (OP_O) stores disconnection acknowledgement (DA) to support its claim to get funds for the services that it provided until roaming occurs. TTP stores the information that this disconnection is due to a roaming to OP_N in order not to get confused when AP_N disconnects without a connection request reached to it.
    - In this scheme, AP_O's signed ticket serves as a formal document, which represents the beginning of the session with AP_N.
4. Client receives $ReqAck$ and forwards it to the new operator (AP_N).
5. AP_N decrypts $ReqAck$ using its private key.
    - AP_N reveals the signed ticket of the AP_O. AP_N sends this signed data to it's affiliated operator to use it for collecting funds from TTP.
    - AP_N verifies the signature over this signed ticket using AP_O's public key. Then, it checks TS in order to decide whether the ticket has expired or not.
    - Then, AP_N starts a challenge-response protocol with the client.
    - The rest of the protocol is the same as Access Point Authentication

Protocol.

7. PAYMENT TO THE OPERATORS (SETTLEMENT)

In the proposed secure and seamless pre-payment scheme, operators claim their money from the TTP by showing their service logs. A log proves a service that has been provided between a connection request and a disconnection request.

$$Log = OpId \,||\, Connection\ Request \,||\, Signed\ Connection\ Response \,||\, TS$$

Operators store connection requests (CR) of the clients; CRs are formed in the Initial Authorization and Reuse of a Connection Card protocols. When a client makes a disconnection request, operator stores the disconnection request (DR) as well. After receiving the DR, operator forms its log as follows.

$$Log = OpId \,||\, Disconnection\ Request\ (DR) \,||\, Signed\ Disconnection\ Response \,||\, TS$$

TS stand for timestamp in the logs. TSs are mandatory in the logs to make TTP's job easier.

When TTP receives two consecutive logs from an operator:

1. TTP will sort the logs according to their TS value.
2. TTP first decrypts CR since it is encrypted with the public key of TTP. CR consists of $Alias$, $Nonce$ and the first hash token to be used to get service.

   Consider

   $$CR = E_{PU-TTP}\ (N \oplus SN \parallel N \parallel H_f)$$

   TTP decrypts it using its private key, and gets SN by the XOR operation:

   $$N \oplus SN \oplus N = SN$$

   Note that SN's first token used is H_f.
3. TTP decrypts the Signed Connection Response using its public key, and gets the alias and the hash token. TTP compares the values with the ones in connection request. If they match, then the log is marked as valid.
4. The abovementioned log is only a service starter; operator needs to show service-ending log to claim its money from the TTP.

   Service ending log naturally has a larger TS value; therefore this log comes later in the sorted list of logs.

   TTP takes the ending log and decrypts DR using its private key.

TTP gets $Alias$, $Nonce$ and the hash token from the decrypted DR. TTP makes the XOR operation: $N \oplus SN \oplus N = SN$ and gets the SN. Note that SN used is the hash token came with the DR to end the service.

5. TTP takes the Signed Disconnection Response and decrypts it using its public key. TTP gets the alias and the hash token from it, and compares the values with the ones came with the DR. If the values match, TTP considers the log as a valid service-ending log.
6. After validating the logs, TTP performs the hash operation over service ending hash token until it reaches the service starter hash token. TTP counts these hash operations. This count is mapped to funds for the provided service.

However the misusage of the logs should be reckoned. Consider the situation of a client:

- Gets service from her home operator between H_0 and H_{10}
- Gets service from a foreign operator between H_{11} and H_{20}
- Gets service from her home operator between H_{21} and H_{30}

In this type of situation home operator has two CRs and DRs, whereas foreign operator has a CR and DR. Home operator has the following logs:

$$Log1 = OpID \,||\, CR_{H_0} \,||\, Signed\ RP_{H_0}$$

$$Log2 = OpID \,||\, DR_{H_{10}} \,||\, Signed\ DA_{H_{10}}$$

$$Log3 = OpID \,||\, CR_{H_{21}} \,||\, Signed\ RP_{H_{21}}$$

$$Log4 = OpID \,||\, DR_{H_{30}} \,||\, Signed\ DA_{H_{30}}$$

The home operator has served between H_0 and H_{10} and also has served between H_{21} and H_{30}. Home operator would want to take the money for serving between H_{11} and H_{20}. It could pretend that it has served the client between H_{11} and H_{20} by not sending $Log2$ and $Log3$. Since $Log2$ indicates that client is disconnected from the operator at H_{10} and $Log3$ suggests that the client started to get service from the operator at H_{21}. Sending only $Log1$ and $Log4$ results TTP to think that the home operator has served the client between H_0 and H_{30}.

This way operator would want money for serving 30 hash tokens.

Abovementioned situation suggests that there should be another operator, which has served between H_{11} and H_{20}. Second operator would have two logs as follows.

$$Log5 = OpID \,||\, CR_{H_{11}} \,||\, Signed\ RP_{H_{11}}$$

$$Log6 = OpID \,||\, DR_{H_{20}} \,||\, Signed\ DA_{H_{20}}$$

Foreign operator proves that it has served between H_{11} and H_{20} by showing the signed RP and DA.

TTP would see that it has already paid home operator for service to that particular client between H_{11} and H_{20}. This means that home operator has tricked TTP to pay more.

In the proposed system TTP is the one who has the authority, it pays operators their money. If the TTP finds an operator misbehaving it could give a penalty to the operator and do not pay for future services, or there could be several other kinds of penalties, since TTP has the proof it could bring the subject to the court as well.

8. SIMULATION ENVIRONMENT

The network topology is hierarchical and WMN supports connections with other IEEE 802.11 protocols [2, 3], clients communicate with TTP via access points, GWs and operators in sequence. Access points are connected to gateways with 6-54 Mbps Wi-Fi connection. Some important specifications about the access points are shown in Table 8.1. *Update Interval* determines the time value between two update packets that access point send to TTP.

The simulator was run on a computer with 2.4 GHz Intel Core 2 Duo, 2 GB 1067 MHz DDR3, Apple MacBook OSX v10.6.8.

Table 8.1: AP Specifications

AP-Gateway Connection bit rate	6-54 Mbps – Wi-Fi
AP-Gateway Distance	100 m
Service Duration per token	5 minutes
Update Interval	11 minutes

The network consists of 32 gateways and 100 access points. In unit simulation there is only one mobile client whereas in real-life scenario simulations there are 300 mobile clients.

Public Key Operations and Their Timings

Public Key Cryptography timings for access points and gateways are mentioned in [33]. For operator servers and TTP servers, timings from [34] are used. For mobile clients, performance values from [35] are used. For AES timings the values from [36] are used, which results a 0.00004 second of delay for AES-128 on Linksys WRT54GS. The same value is used for gateways as well. Timings of hash algorithms are taken from [37] which are considerably lower than symmetric key encryption delay.

Platform specifications are shown in Table 8.2, and RSA-2048 timings are shown in Table 8.3.

Table 8.2: Platform Specifications

	Gateway [11]	Linksys WRT54GS (AP) [11]	Server [12]	Client [13]
CPU Speed	2.08 GHz	200 MHz	Dual-core 64 bit 2.8 GHz	3.2 GHz
CPU type	AMD Athlon XP 2800	Broadcom MIPS32	Intel Xeon	Celeron D 351
RAM	512 MB	32 MB	-	-

Table 8.3: RSA-2048 Timings

	Gateway [11]	Linksys WRT54GS [11]	Server [12]	Client [13]
RSA Signing	1.3 ms	37.9 ms	8.13 ms	1.8 ms
RSA Verification	47.3 ms	1529.0 ms	0.32 ms	-

9. UNIT TEST RESULTS

Unit tests cover protocol behavior under low pressure. In these tests there is only one user, and this user performs the same protocol every minute. These tests are done to ensure that modules of the system are fit for use.

As discussed earlier some protocols show similarity considering packet sizes, cryptographic operations and packet routes. Since there would be no difference between unit tests of protocols that are in the same group, there is one result chart for a particular group of protocols.

9.1 Unit Test Result for End-to-End Two-Way Protocols

Unit tests for end-to-end two-way protocols consist of a user, running the same protocol every minute. Charts present the average delay of packet delivery over time. In this simulation the user sends the packet to a serving access point and the packet hops 2 times in the mesh backbone until it reaches the gateway. Gateway forwards the packet to operator and operator transmits the packet to TTP. TTP processes this packet and sends it back to the client through the same route.

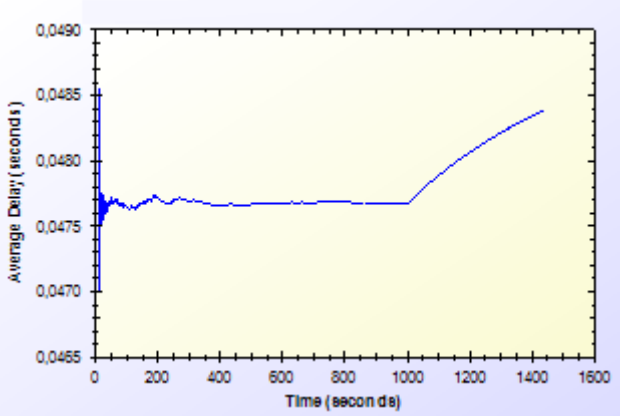

Figure 9.1. End-to-End Two-Way Protocols Unit Test Result

As shown in Figure 9.1, there is a delay that shows variation around 0.04 second. This unstable behavior is caused by different initial packet delays. System needs some packets to set up paths between mesh nodes. The performance stabilizes in time. Average delay shows a peak by the end however the difference between highest and lowest values of the results is

inconsiderable.

9.2 Unit Test Result for Access Point Authentication

Access Point Authentication protocol consists of a challenge-response protocol. It contains two HMAC operations.

Unit test for this protocol contains a user, trying to run access point authentication protocol with a serving access point every minute. The resulting chart, presented on Figure 9.2, shows the average delay of the protocol versus time.

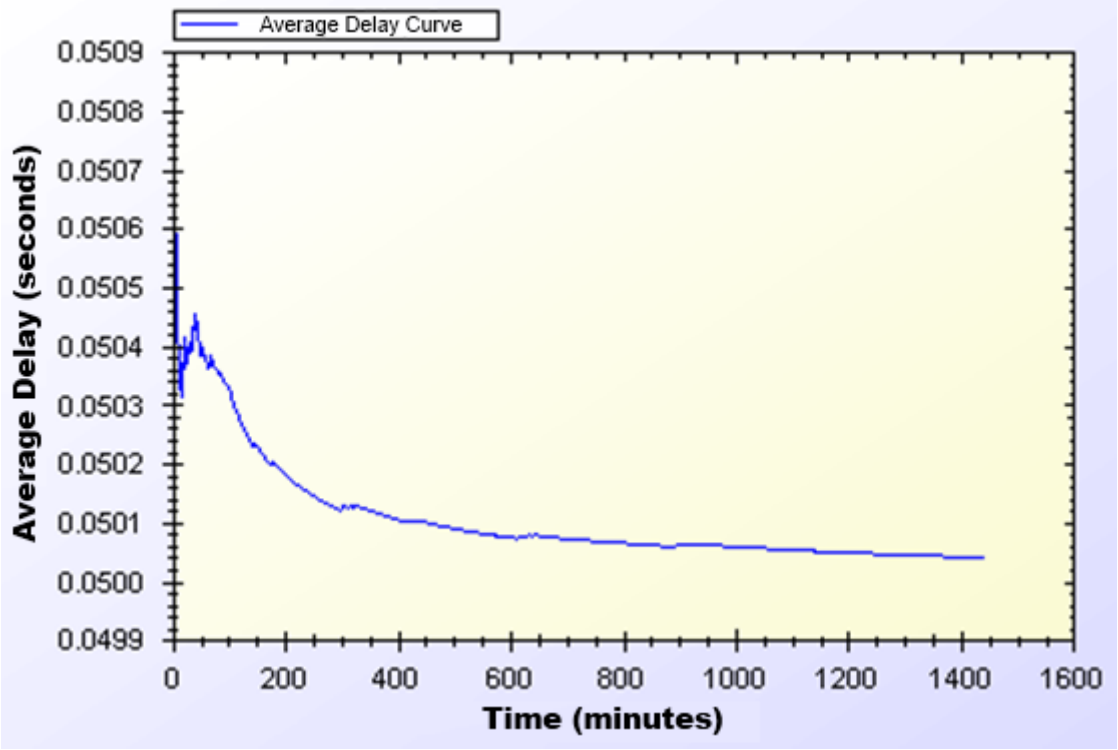

Figure 9.2. Access Point Authentication Protocol Unit Test Result

Average delay of access point authentication converges to 0.05 second in the steady state. The initial delay values are higher than the later ones, because nodes need some time to establish and see who is around. At the time of initial deployment, wireless nodes send and receive beacons and perform operations using them.

9.3 Unit Test Result for Seamless Mobility and Roaming

Seamless Mobility and *Seamless Roaming* protocols have the same behavior since client

sends and receives same length of packets. Thus, they are grouped together for unit tests.

Unit test for *Seamless Mobility* and *Seamless Roaming* protocols consists of a client changes serving access point every minute. Client is located in between two access points and these access points are both eligible for service. Since these protocols must be seamless to the user it is important to get reasonable delays for these protocols.

Figure 9.3 presents the unit test result for *Seamless Mobility* and Roaming protocols.

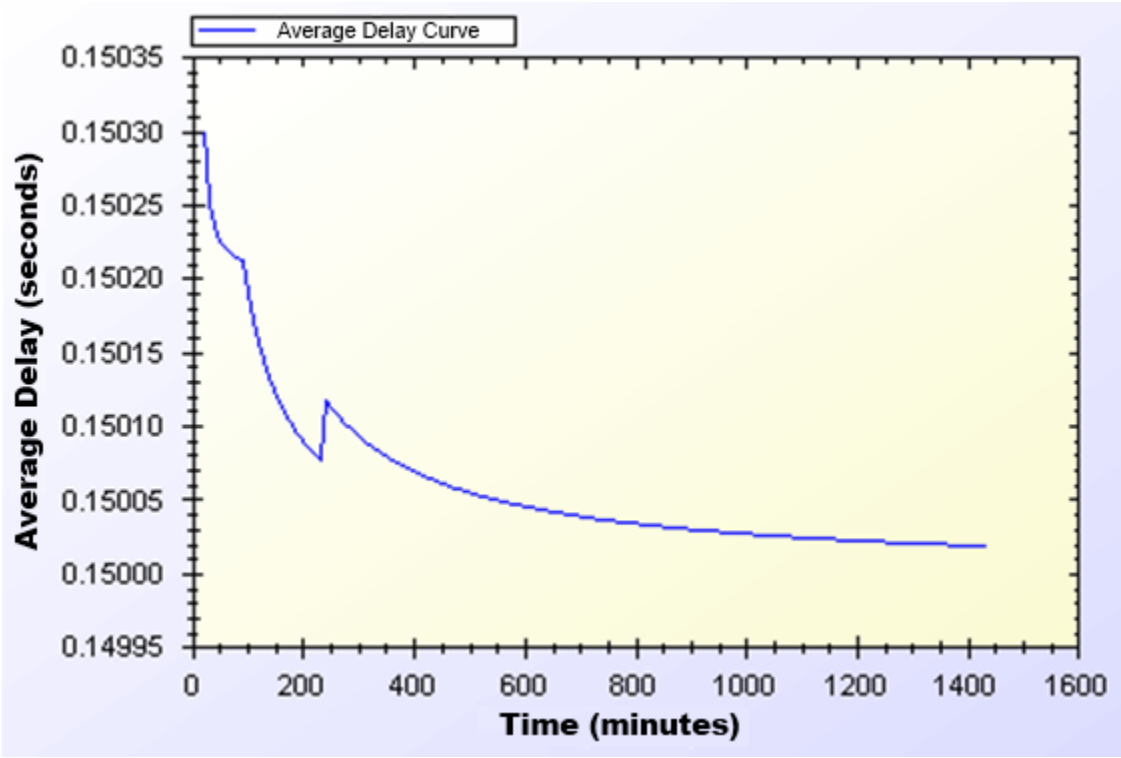

Figure 9.3. Seamless Mobility and Roaming Protocols Unit Test Result

In unit test for these protocols, a 0.15 second of network delay for access point change is observed. Similar to other protocols, there is a transitive period at the beginning of the simulations, however it reaches steady state in time and gains balance.

9.4 Unit Test Result for Packet Transfer

Packet Transfer is the mostly used protocol in the system. It is crucial to have small amount of network delay for this protocol because of it's often use. Unit test scenario of *Packet Transfer* protocol is that a client sends a 512-byte packet every minute.

Figure 9.4 shows the unit test result for Packet Transfer protocol.

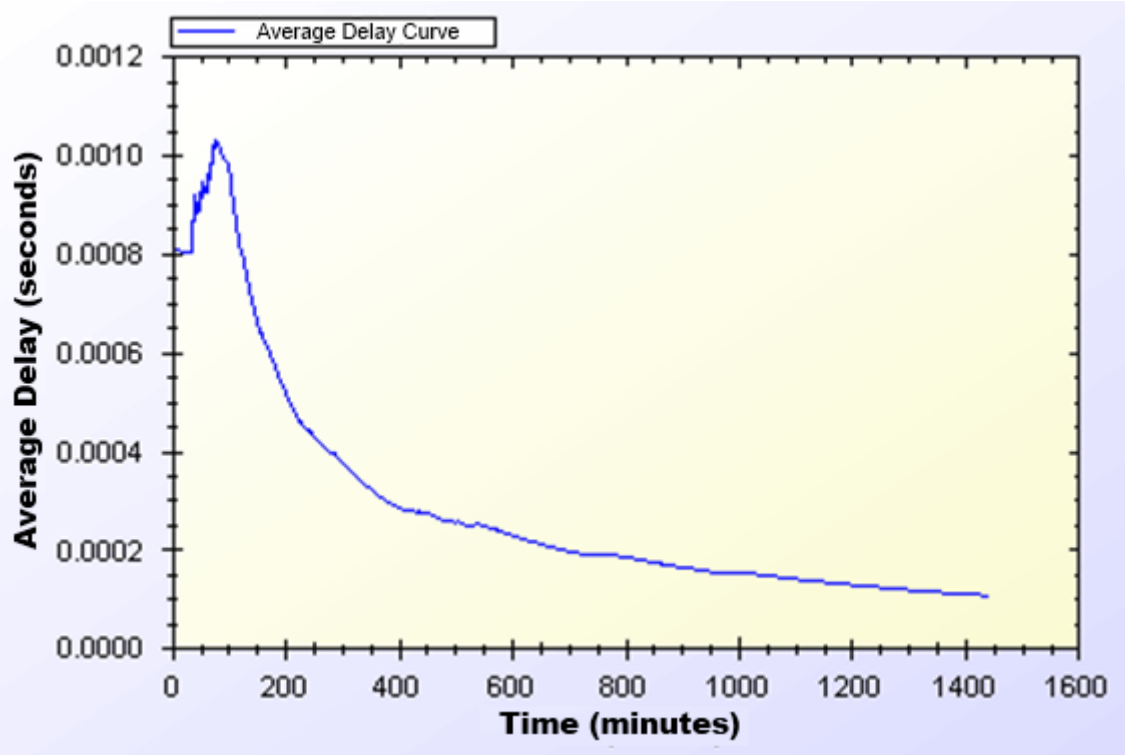

Figure 9.4. Packet Transfer Protocol Unit Test Result

Unit test gave a higher average delay value at the early parts of the simulation but expectedly it reaches a balance through time. As seen on Figure 11, at steady state, packets are received in a very short amount of time, which is around 0.0002 second.

9.5 Unit Test Result for Update Packets

Update Packets protocol takes place between AP and TTP. In this simulation access point updates the user info stored at operator. Figure 9.5 shows the average delay of *Update Packets* protocol over time.

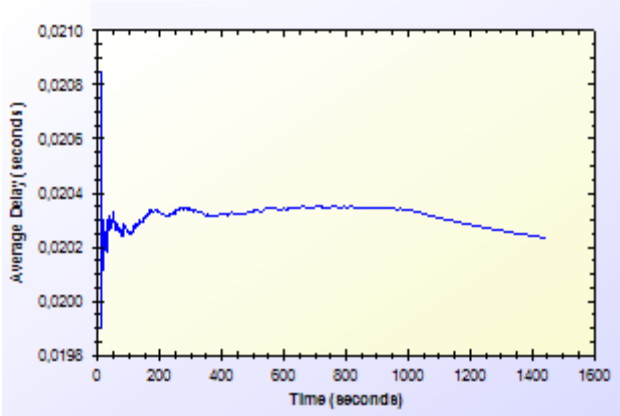

Figure 9.5. Update Packets Protocol Unit Test Result

In the simulation scenario, APs update operator once in every second. Our simulation showed that there is a 0.02 second maximum network delay for updating operator for the client usage.

10.USER MODELING AND MOBILITY

The proposed system intends to serve a variety of users (a.k.a. network clients). Network clients differ in their network usage frequency with respect to time of day, their mobility patterns and frequency of usage.

Certain kinds of actions are defined, such as authorization (initial or reuse of a connection card), disconnection, packet transfer (network usage), payment related roaming and payment related AP handover. All of these actions are triggered as a result of a random event. Connection and network usage related actions are triggered according to a two-state Markov Chain model [8]. Roaming and handoff related actions are triggered by user mobility.

10.1User Actions

In real-life scenario simulations, network usage related actions are modeled using two-state Markov Chain as shown in Figure 10.1. There are two states that a user could be in: *Connected* and *Not Connected*. State transitions or staying in the same state triggers some actions as described below.

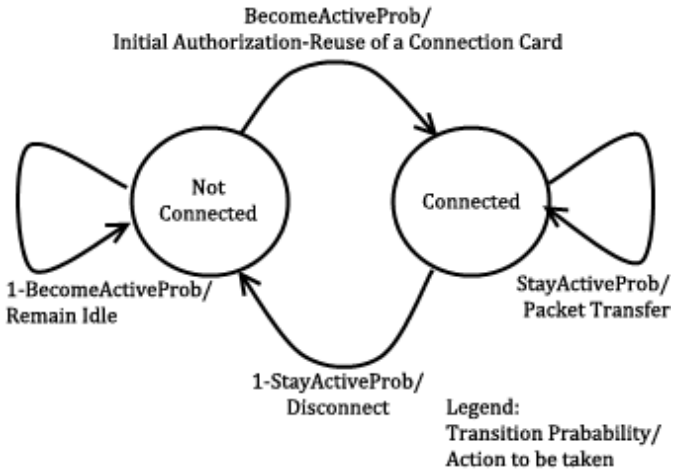

Figure 10.1. State Diagram of Clients

The initial state is *Not Connected*. In this state, the user switches to *Connected* state with the probability value of $BecomeActiveProb$. This state transition triggers *Initial Authorization* (if the CC is used for the first time) or *Reuse of a Connection Card* protocol (if the connection has been used before). In this way, the user starts consuming the network and

getting the service. While in *Not Connected* state, the user stays in the same state with probability value of $1 - BecomeActiveProb$.

While in *Connected* state, the user remains connected (i.e. stay in the same state) with the probability of $StayActiveProb$. Staying connected triggers *Packet Transfer* protocol. In other words, the user continues to get service via the currently connected AP. In *Connected* state, transition to *Not Connected* state occurs with probability of $1 - StayActiveProb$. This transition disconnects the user via *Disconnection* protocol.

In this 2-state Markov chain model, the average connection duration, T_{con}, is calculated as the expected value of staying in *Connected* state, as given below.

$$T_{con} = \sum_{i=1}^{\infty}(1 - P_{SA}) \cdot i \cdot {P_{SA}}^{i-1} = (1 - P_{SA}) \sum_{i=1}^{\infty} i \cdot {P_{SA}}^{i-1} = \frac{1}{1-P_{SA}} \quad (1)$$

Where, P_{SA} denotes $StayActiveProb$.

The expected value of staying in *Not Connected* state is the average idle time for a user between two connections. This value, T_{idle}, is calculated as follows.

$$T_{idle} = \sum_{i=1}^{\infty} P_{BA} \cdot i \cdot (1 - P_{BA})^{i-1} = P_{BA} \sum_{i=1}^{\infty} i \cdot (1 - P_{BA})^{i-1} = \frac{1}{1-(1-P_{BA})} = \frac{1}{P_{BA}} (2)$$

Where, P_{BA} denotes $BecomeActiveProb$.

10.2Client Types

Three different user types are outlined with different networking and mobility requirements. Considering whether they are working, studying or domestic provides the differentiation among user types.

The network usage within one day has been modeled in three time slots: (i) night (00:00 – 07:59), (ii) daytime (08:00 – 15:59), and (iii) evening (16:00 – 23:59).

User types are described as follows:

- **Students:** This kind of clients uses network services mostly in the evening when they return back from school. Their possibility to use network services during morning and night is relatively small comparing to mid-day time. Thus, the

probabilities for being active are higher for evening. Students are assumed to be mobile at the beginning and end of the *daytime* slot since they go to their school. Until the end of the *night* slot, students would more likely to get service in their homes in an immobile way.

- **Employees:** This kind of clients has routine lives. They are immobile and not so active during nights. However, during the daytime, they are very active and use network services at their work places. Moreover, they are mobile as they commute to/from work from/to home at the beginning and end of the working times.

- **Domestics:** This type of users does not work outside and spend their time at home. Usually the domestics get Internet service in an immobile way. These users are highly active at all times.

The parameters of $StayActiveProb$ and $BecomeActiveProb$ are determined based on the abovementioned discussion about the client type characteristics and the time slots. These values are given below. The triplet $\{x, y, z\}$ specify the probability values for night, daytime and evening, respectively.

$$becomeActiveProb < Domestic > = \{0.40, 0.60, 0.60\};$$

$$becomeActiveProb < Student > = \{0.20, 0.20, 0.80\};$$

$$becomeActiveProb < Employee > = \{0.20, 0.99, 0.20\};$$

$$stayActiveProb < Domestic > = \{0.90, 0.98, 0.80\};$$

$$stayActiveProb < Student > = \{0.30, 0.20, 0.98\};$$

$$stayActiveProb < Employee > = \{0.30, 0.99, 0.20\};$$

These values also determine the average connection duration and idle time by using Eq. 1 and 2. For example, a domestic client remains idle during daytime for $\frac{1}{1-(1-0.6)} = \frac{1}{0.6} = 1.67$ minutes between connections. Once connected, average connection time for this

category is $\frac{1}{1-0.98} = \frac{1}{0.02} = 50$ minutes.

10.3User Mobility and Timing

Real-time scenario covers Internet usage of 300 users in a 1-km^2 metropolitan area. The simulations time begins at 00:00 a.m. and lasts for 24 hours. Simulation time is divided into 3 parts considering night, daytime and evening. Every part of the day has different statistical values for client behaviors.

Simulations are run for 1440 seconds, however every second in the simulation stands for 1 minute in real life.

In real-life scenario simulations clients are able to move from one location to another. The time and direction of their movement is selected at random but probabilities are affected by user roles. For example, when school is over, a student is most likely to move towards her target destination (e.g. her home).

Clients are assigned a random target access point. Every one of 100 access points has 3 initial clients. The client moves from its current access point to the target access point on the grid. An example movement pattern is shown in Figure 10.2. As a client moves from access point A to the access points B, if she needs to connect to the Internet, she forms up a new connection with the access point, which is closest to client's current location.

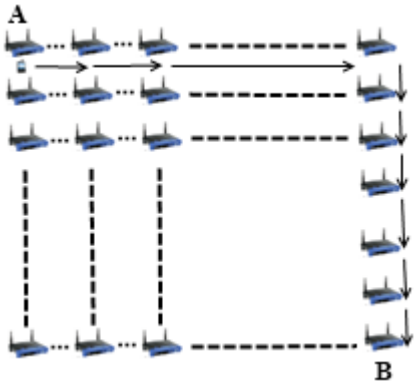

Figure 10.2. User Movement from A to B

In real-life scenario simulations, there are two operators and they have same amount of access points. In current simulations, each operator has 50 access points. The client executes

handover or roaming if there is an active connection during movement between access points. In such a case, depending on the new access point's affiliated operator, user's movement triggers either *Seamless Mobility* or *Roaming* protocols. If new access point's affiliated operator is same as the one that client currently uses, and then it means the client would perform *Seamless Mobility* protocol for handover. Otherwise, the client would run *Seamless Roaming* protocol.

Clients are assigned uniformly distributed random speeds between 2 km/h to 6 km/h. The clients are assumed to move without a motor vehicle.

11.RESULTS FOR REAL-LIFE SCENARIO SIMULATION

Results for unit test simulations are described before; however the most significant results are real-life scenario simulation results. Despite the randomness of the system, users' actions are highly related to their group and current simulation time.

Charts for the results display the average delay for a particular protocol.

9.6 Overview

Final simulations provided the results in Table 11.1. Charts on Figure 11.1 and Figure 11.2 are drawn exploiting the results in Table 11.1. Considering the results it could be calculated that over 100 minutes of Internet service, workers have only waited for 1 minute for system delays. In average, over 1000 minutes of Internet service needs a delay of 13 to 16 minutes of waiting.

Table 11.1 Simulation Results for Client Types

	Total Internet Usage Time	Total Internet Usage Delay	Average Internet Usage Time for a Client	Average Internet Usage Delay for a Client
Student	95899,26 Minutes	1698,95 Minutes	958,99 Minutes	16,98 Minutes
Worker	101681,64 Minutes	1316,35 Minutes	1016,81 Minutes	13,16 Minutes
Non-Worker	105335,08 Minutes	1456,12 Minutes	1053,35 Minutes	14,56 Minutes

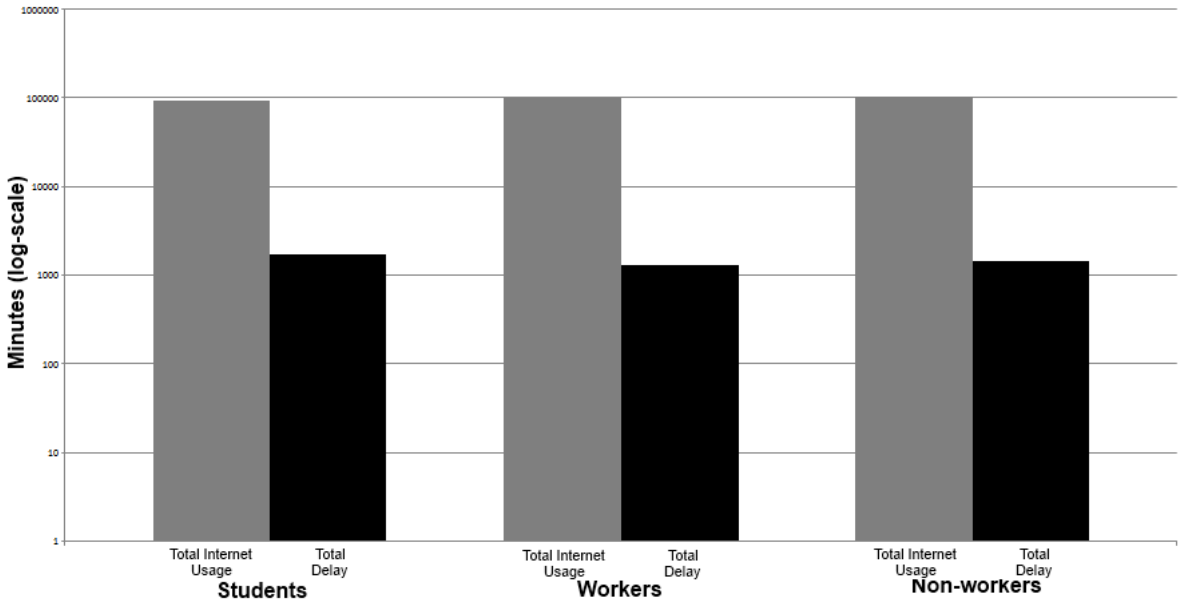

Figure 11.1. Total Amount of Service Usage Times for Client Types vs. Total Delays

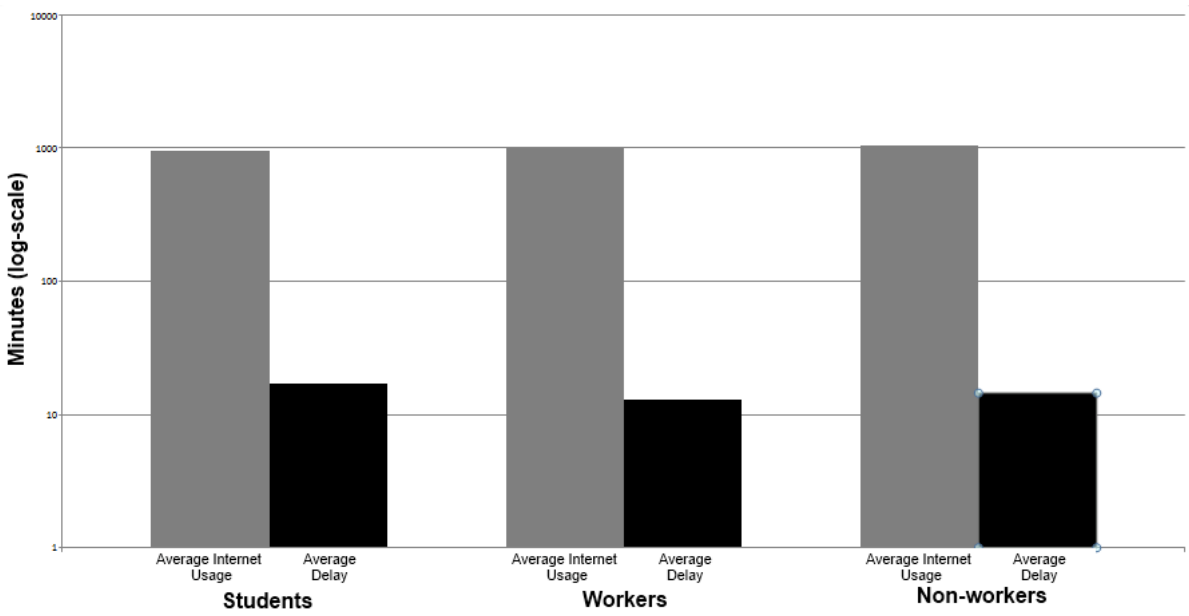

Figure 11.2. Average Service Usage Times for Client Types vs. Average Delays

As described before the clients are grouped into 3 groups. The client roles and probabilistic values affect their behavior in the system, which results difference between overall values of the simulations.

Figure 11.1 and Figure 11.2 shows the overall results for real-life scenario simulation. Figure 11.1 shows comparison of minutes clients used as idle or active. Figure 11.2 shows the average value for the clients of the same group.

9.7 Real-Life Scenario Simulation Result for Initial Authorization

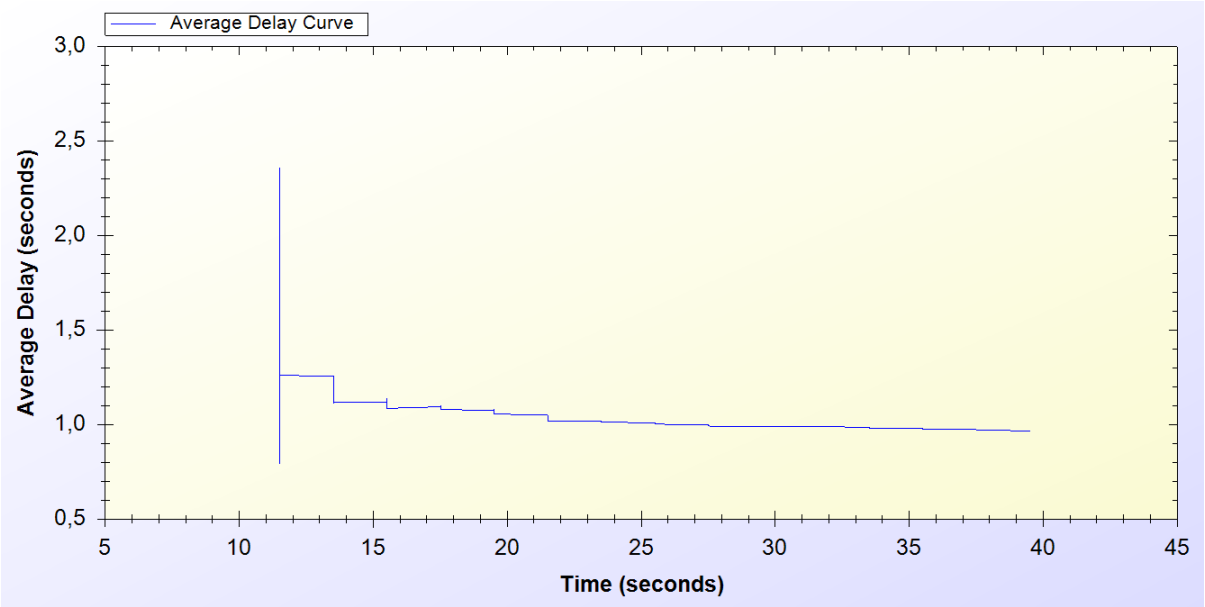

Figure 11.3. Real-life Simulation Result for Initial Authorization Protocol

Initial Authorization protocol is used at the beginning of the service for each user. As it is seen on the chart every one of the 300 users are authenticated at the end of 40th minute.

Simulation starts around the 10th minute in the morning. At the beginning there is a huge amount of users, trying to authenticate. Figure 11.3 indicates that, this process varies between 0.6 and 2.5 seconds. After 10 minutes it attains a balance and *Initial Authorization* protocol meets a delay of 1 second, which means when users open up their mobile device they would have Internet service after 1 second.

9.8 Real-Life Scenario Simulation Result for Reuse of a Connection Card Protocol

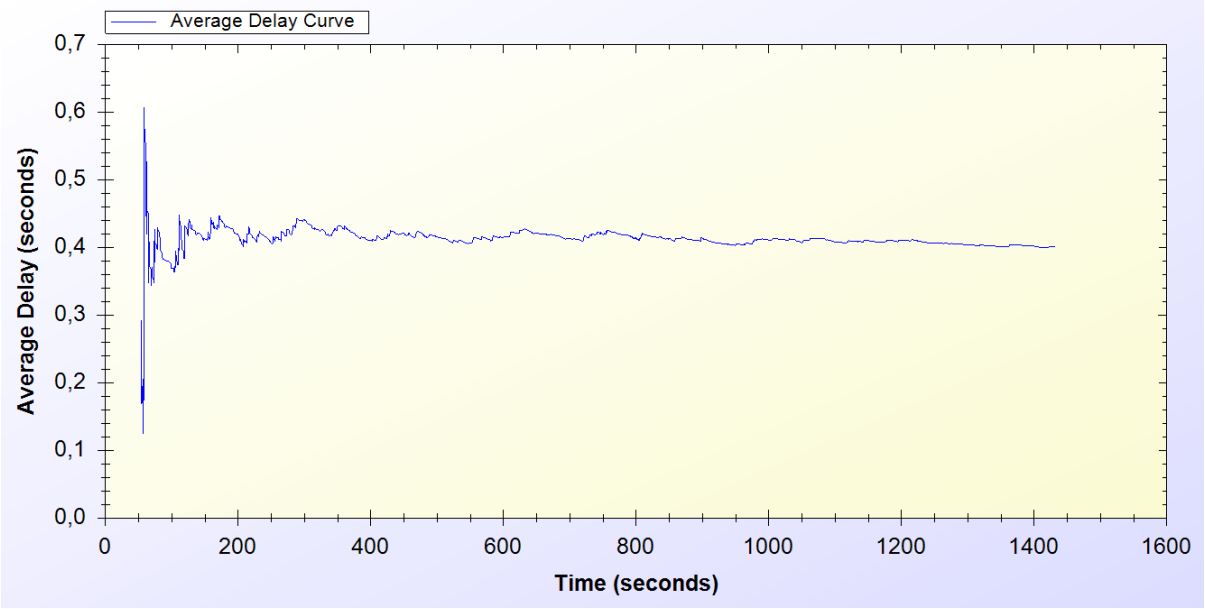

Figure 11.4. Real-Life Simulation Result for Reuse of a Connection Card Protocol

Reuse of a Connection Card protocol is used after disconnecting from the system. As it is seen it is a highly used protocol in the system. It starts around the 50th minute and used for the entire time of the simulation.

As seen on Figure 11.4, at the beginning of the protocol the delay changes between 0.1 and 0.6 second. After some time protocol achieves a balance and a 0.4 second of network delay is observed.

9.9 Real-Life Scenario Simulation Result for Changing Alias

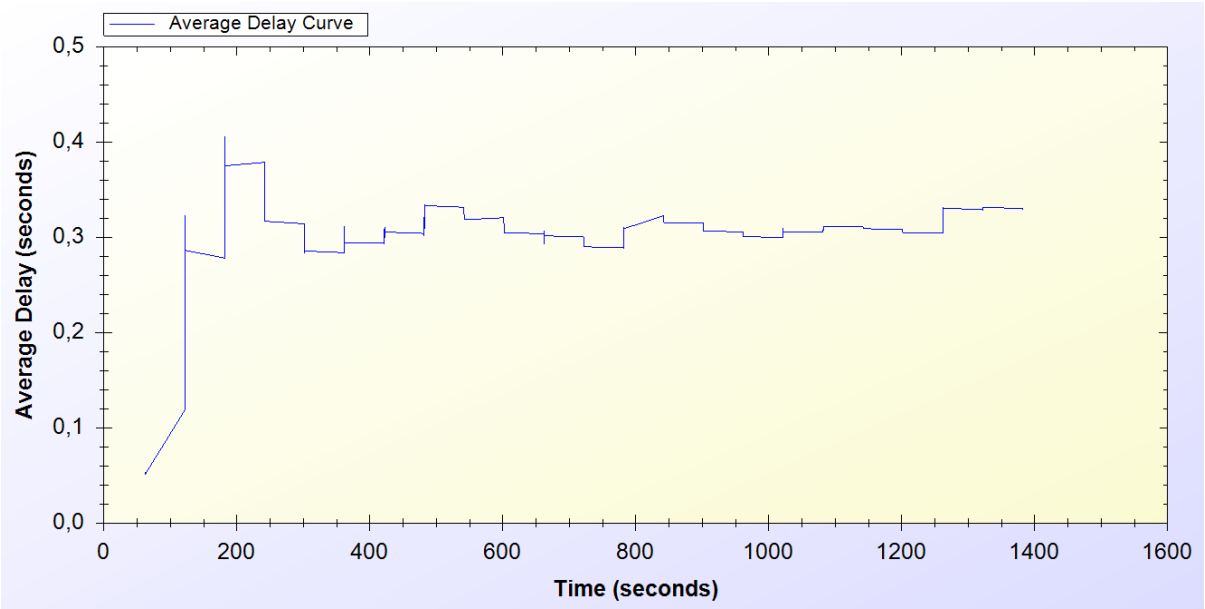

Figure 11.5. Real-Life Simulation Result for Changing Alias Protocol

Every active client uses *Changing Alias* protocol in the system in every 50 minutes. The protocol is first used at 50th minute and it is used entire time of the simulation.

As one can see on Figure 11.5, at the beginning of the protocol the delay for the protocol varies between 0.1 and 0.4 seconds. After some time the average delay for the protocol converges to 0.4 seconds.

9.10 Real-Life Scenario Simulation Result for Disconnection

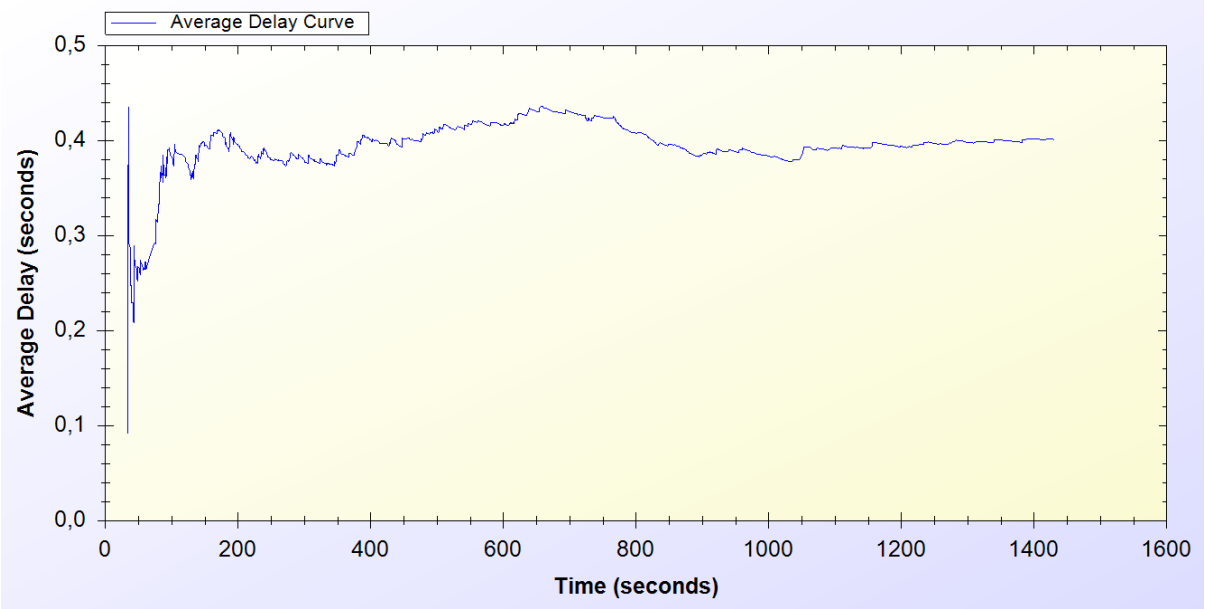

Figure 11.6. Real-Life Simulation Result for Disconnection Protocol

Disconnection protocol first appears around 30th minute and it is used through the entire time of the simulation. Figure 11.6 shows that, at the beginning of the system Disconnection protocol average delay vary between 0.1 and 0.5 second but through time the average delay meets 0.4 second.

9.11 Real-Life Scenario Simulation Result for Update Packets

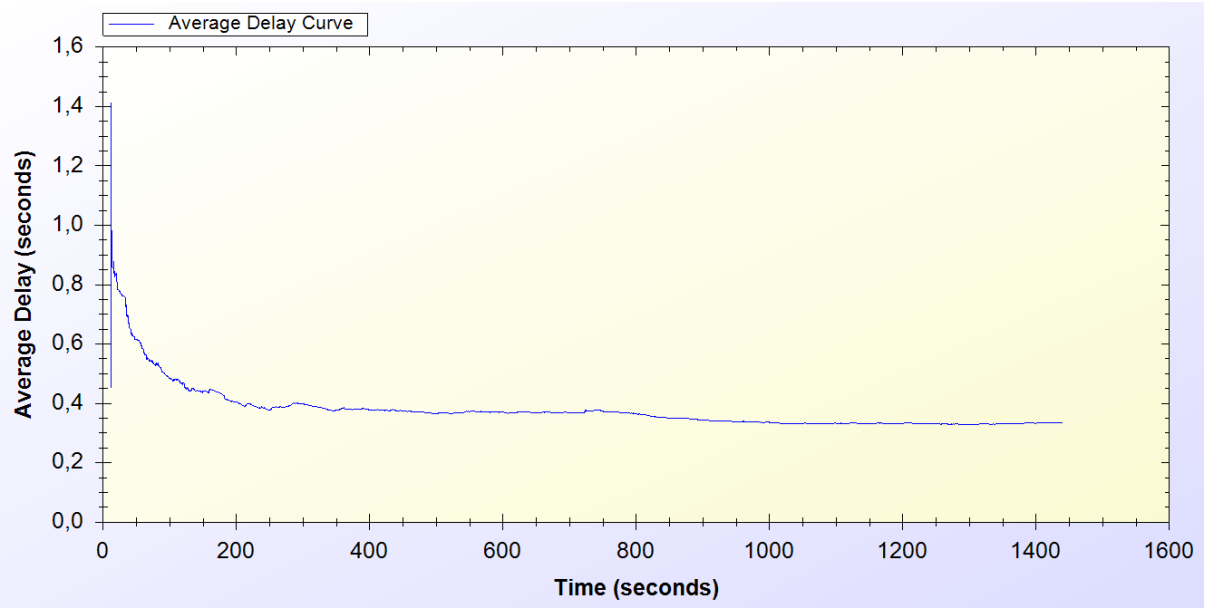

Figure 11.7. Real-Life Simulation Result for Update Packets

Update Packets protocol is an end-to-end one-way protocol. It is expected to get lower delay values for this one. Only access points use *Update Packets* protocol and they send packets to TTP. The packets are sent every 10 minutes.

As it is seen on Figure 11.7, at the early stages of the protocol, the average delay value varies between 0.6 and 1.4 second but then after some time the protocol stabilized around 0.4 second.

9.12 Real-Life Scenario Simulation Result for Seamless Mobility in Home Operator Protocol

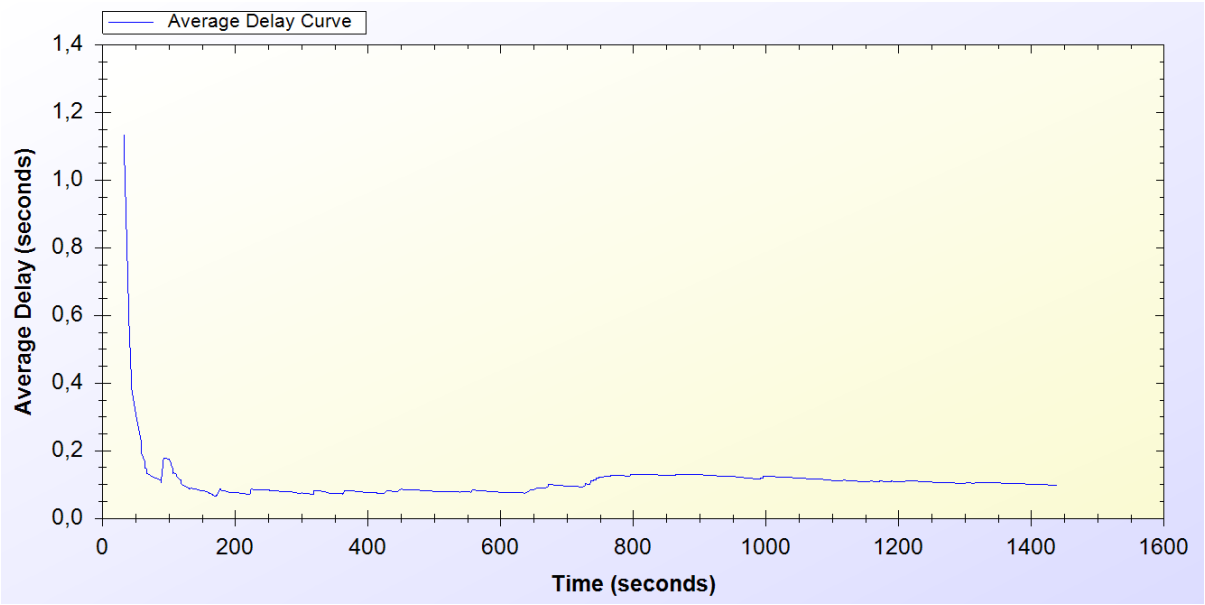

Figure 11.8. Real-Life Simulation Result for Seamless Mobility Protocol

Seamless Mobility protocol is used when a handover happens between access points. If these access points are belonging to the same operator then it means the client is using *Seamless Mobility* protocol.

By looking at Figure 11.8, it could be said that, *Seamless Mobility* protocol has an initial average delay that shows difference between 0.2 and 1.2 seconds. A user loses around 0.1 second to make a handover to the new access point.

9.13 Real-Life Scenario Simulation Result for Roaming Protocol

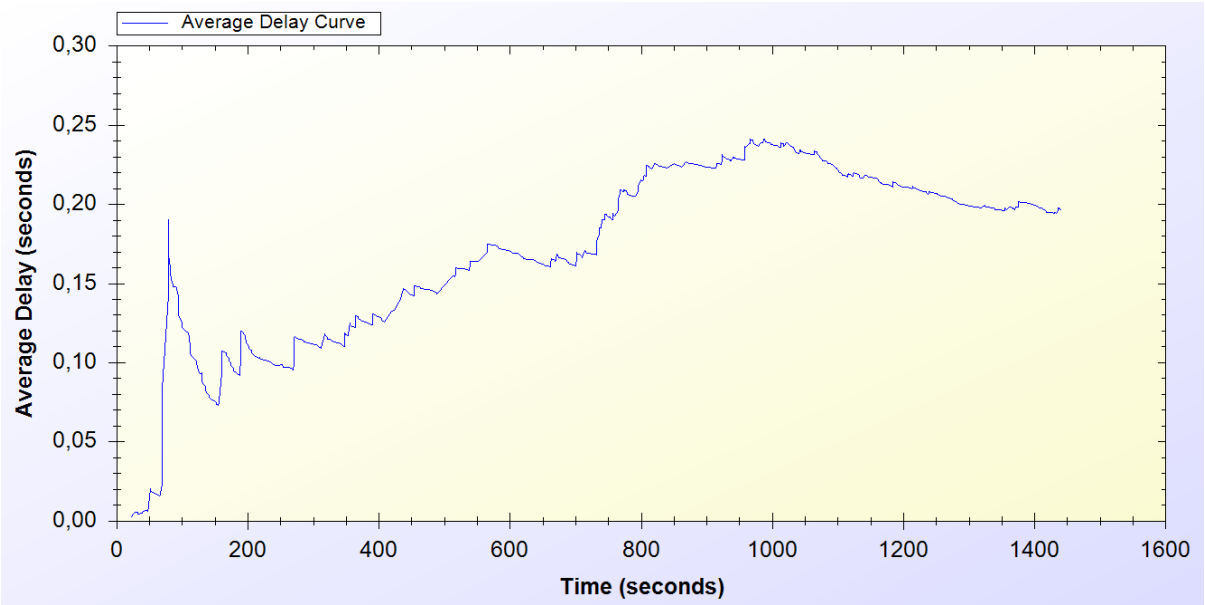

Figure 11.9. Real-Life Simulation Result for Roaming Protocol

Roaming protocol is used when a handover happens between access points. If these access points are belongings of different operators then it means the client is using *Roaming* protocol.

Roaming protocol has an average delay that varies between 0.05 and 0.2 seconds. There are 2 operators so a client has a %50 chances to make a *Seamless Mobility* or *Roaming* protocols. After some time protocol reaches a balance around 0.2 second of delay.

As one can see on Figure 11.9, the results for *Roaming* protocol shows a boost until the middle of the simulation time but it decreases and achieves a balance

9.14 Real-Life Scenario Simulation Result for Packet Transfer

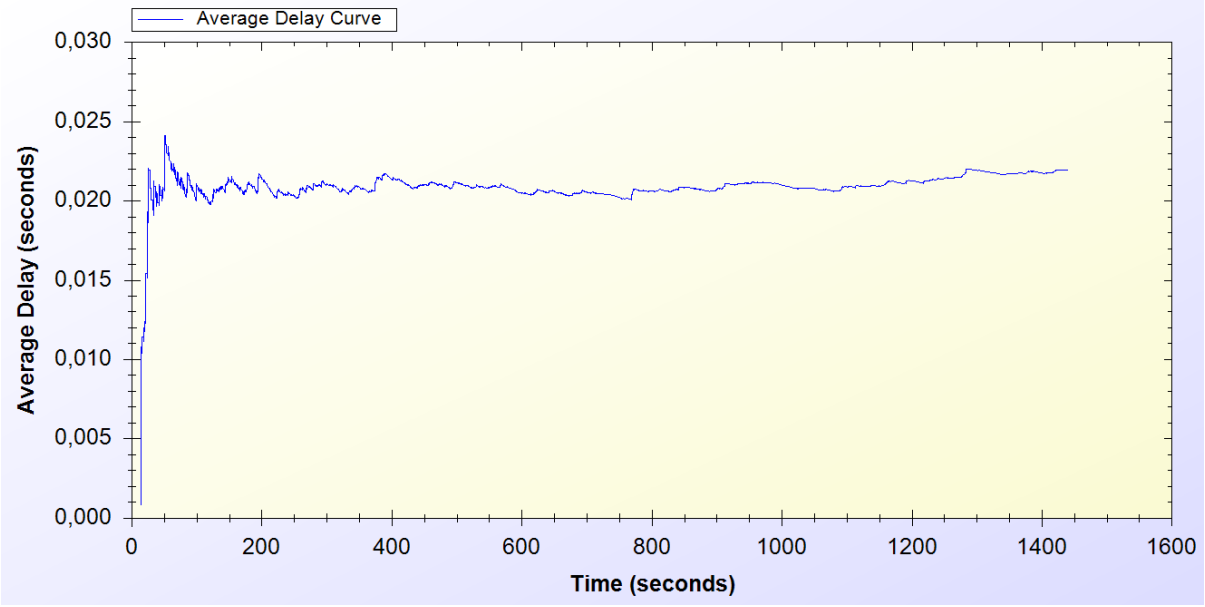

Figure 11.10. Real-Life Simulation Result for Packet Transfer Protocol

Packet Transfer protocol is the mostly used protocol in the system.

Figure 11.10 states that, at the beginning of the protocol the average delay value varies between 0.005 and 0.025 but then the protocol achieves a balance around 0.02 second.

12.DISCUSSION

In section 4 the requirements for a secure and seamless pre-payment scheme were discussed. In this section the success of the proposed system on meeting the requirements and simulation results are discussed.

Wide Coverage: Every access point could serve within a 100 meters radius. It is proven that with 100 access points, 1 km^2 area is covered for Internet service.

Seamless Connection and Roaming: Users are able to switch between access points no matter what operator they belong to. The delays are low enough to maintain current connection without any interruption.

Roaming/mobility: Reuse of a connection card is possible after attempting first connection. Roaming is supported, when our protocol is implemented in participating *AP*s, and tokens are valid.

Anonymity: For legal purposes users must give their identities to connection card issuer (*TTP*) for getting connection cards. Therefore, as far as *TTP* keeps clients' identities secret, users can stay anonymous. However, every action of the client is completed using their aliases.

Mutual authentication: Initial Authorization and Reuse-CC protocols ensure the authentication of the user. TTP signs the acknowledgement values and a handshake protocol is run between the serving access point and the client. These processes ensure mutual Authentication.

Two-way honesty: Since the tokens are issued by *TTP*, only the *TTP* and connection cardholder knows all the tokens that are related with a specific connection card. Hence whenever a Client sends a new token, it is not possible for him to say "I did not use it". Since *TTP* is a trusted third party, in the roaming phase, operators cannot say that they provided service for non-used tokens.

Preventing double spending: All the connection card information is stored in the database with *In Use* field. Therefore it is not possible for two users to use the same connection card at the same time. Since the last token information is stored in the database, it

is not possible to double-spend a token.

Untraceability: Our protocol provides untraceability by changing aliases periodically. Clients are traceable between the times they change their aliases nonetheless they could not be related to future actions after the alias change. The period of time to change the aliases is a choice of the designer.

Performance: The performance of the system is tested using ns-3. Both unit tests and real-life scenario results are obtained. In both simulations system protocols achieved a steady state performance. Achieving stable performance is significant because it represents the system behavior in rush situations and long run.

13.CONCLUSION

In this thesis we have proposed a secure and seamless pre-payment scheme for network service. The system uses outputs of hash operations in reverse manner to use irreversibility property of hash functions. Fast computation of hash algorithms ensures a fast and secure system for Internet service pre-payment.

In unit tests, standalone performances of the protocols under trivial usage scenarios are analyzed. Unit tests set an example for how the system will behave in empty hours. In this way, the first proof-of-concept implementation of the system is provided and showed that the designed protocols reach steady state.

In real-life scenario simulations we have tested the proposed scheme using complex scenarios with realistic client types and movement patterns. Expected behavior of SSPayWMN is took form after these simulations since they are most close-to-real-life simulations of all the SSPayWMN simulations performed before.

Uniform probability distribution model enabled us to simulate real time scenarios in simulation environment, and gets results closer to real time situations. Different client types are used to make simulations closer to real-life. There is also randomness in the system, so there occurred different outcomes for the same simulation.

Unit tests and real-life scenario simulation results show that the proposed system is a considerable and an effective pre-payment system.

14.REFERENCES

[1] Akyildiz, I. F., Wang, X., and Wang, W. (2005) Wireless mesh networks: a survey, *Computer Networks and ISDN Systems*, 47(4): 445-487.

[2] Intel Inc., Multi-Hop Mesh Networks—a new kind of Wi-Fi network.

[3] J. Walker, Wi-Fi mesh networks, the path to mobile ad-hoc. Available from http://www.wi-fitechnology.com/ Wi-Fi_Reports_and_Papers/Mesh_Networks_References.html.

[4] Vaughan-Nichols S.J., (2004) Achieving wireless broadband with WiMax, *IEEE Computer*, vol. 37, no.6, pp. 10-13.

[5] The ZigBee Alliance. Available from: http://www.zigbee.org.

[6] Network Simulator 3 (ns-3) http://www.nsnam.org

[7] Trappe, W. Washington, L. C. Introduction to Cryptography with Coding Theory, Second Edition, pp. 218-220

[8] Oligeri, G. Chessa, S. Giunta, G. Loss Tolerant Video Streaming Authentication in Heterogeneous Wireless Networks, Computer Communications, 34(11): 1307-1315, 2011.

[9] American Bankers Association, *Keyed Hash Message Authentication Code*, ANSI X9.71, Washington, D.C., 2000

[10] H. Krawczyk, M. Bellare, and R. Canetti, *HMAC: Keyed-Hashing for Message Authentication*, Internet Engineering Task Force, Request for Comments (RFC) 2104, and February 1997.

[11] FIPS PUB 46-3 (1999) Data Encryption Standard (DES), http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf

[12] FIPS PUB 197 (2001) Announcing the Advanced Encryption Standard (AES), http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[13] Biryukov A. and Kushilevitz E. (1998). Improved Cryptanalysis of RC5. *EUROCRYPT* 1998.

[14] Bruce Schneier (1993) Description of a New Variable –Length Key, 64 bit Block Cipher (Blowfish), http://www.schneier.com/paper-blowfish-fse.html

[15] Robshaw, M. J. B. (1995) Stream ciphers, RSA Laboratories Technical Report.

[16] W. Stallings, Cryptography and Network Security: Principles and Practices, 3rd edition, Prentice Hall, NJ, 2003.

[17] Managed File Transfer and Network Solutions, http://www.jscape.com/blog/bid/82975/Which-Works-Best-for-Encrypted-File-Transfers-RSA-or-DSA

[18] Coleridge, R. (1996) The Cryptography API, or How To Keep A Secret, *Microsoft Security Technical Articles*, http://msdn.microsoft.com/en-us/library/ms867086.aspx

[19] RFC 2631—Diffie-Hellman Key Agreement Method E. Rescorla June 1999.

[20] FIPS PUB 186-3 (1994) Digital Signature Standard (DSS) - CSRC, http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf

[21] Elgamal, T. (1985) A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, *IEEE Transactions on Information Theory*, http://caislab.kaist.ac.kr/lecture/2010/spring/cs548/basic/B02.pdf

[22] Rivest, R., Shamir, A., and Adleman, L. (1978) A method for obtaining digital signatures and public-key cryptosystems, *Communications of the ACM*,21(2): 120–126.

[23] FIPS 180-3 (2008) Secure Hash Standard (SHS), http://csrc.nist.gov/publications/fips/fips180-3/fips180-3_final.pdf

[24] RFC 1321 (1992) The MD5 Message Digest Algorithm, http://tools.ietf.org/pdf/rfc1321.pdf

[25] Joseph D. Camp and Edward W. Knightly The IEEE 802.11s Extended Service Set Mesh Networking Standard

[26] Kai Yang, Jian-feng Ma, Zi-hui Miao (2009) Hybrid routing protocol for wireless mesh network, Computational Intelligence and Security – CIS '09

[27] Ethernet Prototype Circuit Board, Smithsonian National Museum of American History, Retrieved 2007-09-02.

[28] Chaum, D. (1982) Untraceable electronic mail, return addresses, and digital pseudonyms, *Communications of the ACM, 4(2).*

[29] Rao, Y.S.; Wing-Cheong Yeung; Kripalani, A., "Third-generation (3G) radio access standards," *Communication Technology Proceedings, 2000. WCC - ICCT 2000 International Conference on*, vol.2, no., pp.1017-1023 vol.2, 2000 doi: 10.1109/ICCT.2000.890849

[30] Zaghloul, S., Bziuk, W. and Jukan, A. "A scalable Billing Architecture for Future Wireless Mesh Backhauls", IEEE ICC '08.

[31] Zhang, Y. and Fang, Y., "A secure authentication and billing architecture for wireless mesh networks", Wireless Networks, vol.13, no. 5, pp. 663-678, October 2007.

[32] Lamport, L. (1981) Password authentication with insecure communication, *Proceedings of Commun. ACM,* vol. 24, no. 11,pp. 770-772.

[33] Efstathiou, E., Frangoudis,P., and Polyzos,G. (2006) Stimulating Participation in Wireless Community Networks, *IEEE INFOCOM, 2006,* Barcelona, Spain.

[34] Deng, L., and Kuzmanovic, A., (2009) A feeder-carrier-based internet user accountability service, *Northwestern University Technical Report,*http://networks.cs.northwestern.edu/susinet/TR-09-12.pdf

[35] Yakovyna, V., Fedasyuk, D., Seniv M., Bilas O. (2007) The performance testing of RSA algorithm software realization, *CAD Systems in Microelectronics*, *CADSM '07*, pp. 390-392, Polyana, UKRAINE.

[36] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, N. Ferguson Performance Comparison of the AES Submissions, Proc. Second AES Candidate Conference, NIST, March 1999, 15-34.

[37] Crypto++ 5.6.0 Benchmarks: http://www.cryptopp.com/benchmarks.html

[38] Yucel, C. (2010). A Secure Prepaid Micropayment Scheme for Wireless Mesh Networks (Unpublished master's thesis). Sabanci University, Istanbul, TR.

[39] David B. Johnson and David A. Maltz. Dynamic source routing in ad hoc wirelees networks. In Mobile Computing, edited by Tomasz Imielinski and Hank Korth chapter 5, pages 153-181. Kluwer Academic Publishers, 1996.

[40] OMNET++ Discrete Event Simulator, http://www.omnetpp.org

[41] P. Crescenzi, M. Di Ianni, A. Marino, G. Rossi, and P. Vocca. Spatial Node Distribution of Manhattan Path Based Random Waypoint Mobility Models with Applications. In Sirocco 2009, to appear.