Secure and Seamless Payment for Wireless Mesh Networks

Serhat Can Leloğlu, Can Yücel, Albert Levi

Sabanci University, Turkey

canleloglu@sabanciuniv.edu

canyucel@sabanciuniv.edu

levi@sabanciuniv.edu

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. In this paper 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. Additionally SSPayWMN protects clients' anonymity and provides unlinkability for the client actions. The implementation of the system is made on a network simulator and simulation results are presented in this paper. SSPayWMN has achieved remarkable results in the simulations; system protocols reached steady state in every simulation, which ensures the stability of the system.**

***Keywords*—Wireless Mesh Networks, Cryptography, Payment Systems, Security, Network Simulation**

I. 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 [18], ZigBee [19] and 3G-radio access [20] could also inter-connect with WMN structure.

There has been research for developing secure pre-payment systems for Internet access. In [8], 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 [9], 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 [8] 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 unlinkability. 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 employs some cryptographic primitives to ensure system security. The billing system counts on hash chains [10] 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 behaviour 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.

The rest of the paper is organized as follows: First we give a brief overview for SSPayWMN and suggested network topology in Section 2. In Section 3 we explain the system protocols. The settlement of the operators and the money transfer is explained in Section 4. Simulation environment is explained in Section 5 and unit test results are presented in Section 6. We give brief explanation for user modelling and mobility in Section 7. Simulation results for real-life scenario are presented in Section 8. A discussion on system success and properties is in Section9. Finally conclusion is given in Section 10.

II. General Overview of Proposed Scheme and System Entities

The proposed system is a secure pre-payment infrastructure for WMNs that also considers users' privacy and fairness. In this infrastructure there are mobile phones or laptops as clients, as well as tools that are used for service providing. Table 1 gives a list of system entities that function in the proposed system.

TABLE I
SYSTEM ENTITIES

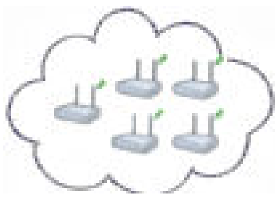
	Mobile user (client)
	Access Point (AP). From now on in this document, it is called as AP, but please note that it also has routing capability.
	Mesh backbone
	Gateway (GW) that connects the mesh backbone to outer world and also to the operator's server
	Operator's server (OP). Keeps necessary logs and user info.
	Trusted Third Party (TTP). Payment related logs are mostly to be generated by the TTP.

Figure 1 shows the topology of the network and connections between entities.

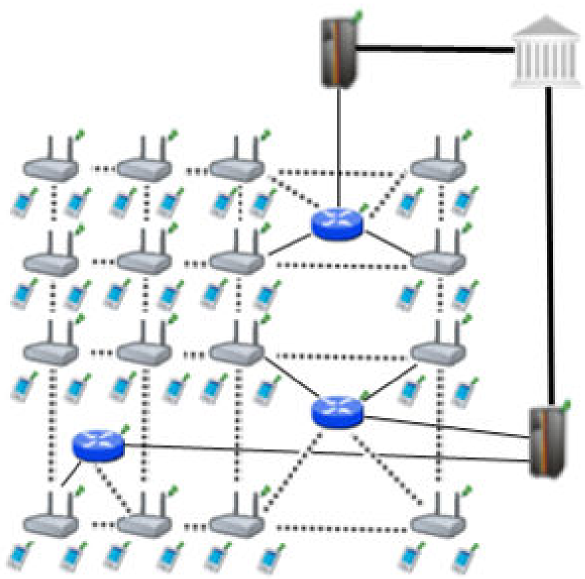


Figure 1. Network Topology

Connection between serving access points is wireless, and they use IEEE 802.11s protocol [6]. 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) [7], which is a hybrid routing protocol, which has routing tables.

Connection medium between mesh backbone and gateway (GW) is wireless. GWs and operators communicate through wired connection. The connection between an operator and TTP is also wired. These connections use 802.3(Ethernet protocol) [17].

A. Connection Card Structure

A Connection Card (CC) is the main deed that clients buy from operators and use to get Internet service. CCs include credits as tokens. Hash tokens are generated using hash chains as discussed below. CCs also have unique Serial Numbers (SN), which are to be used for alias computation later.

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 several times. The number of hash operations is actually the number of tokens in a set.

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

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

$$\dots$$

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

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

H_0 is the first token to be used, then tokens are used in increasing order by token index. In this manner, one-way property of hash algorithms is exploited such that an attacker cannot learn the next token even if she knows the previous tokens.

B. 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 3.

C. Notations

The symbols and operators used in this paper are listed in Table 2.

TABLE II

System Entities

$\oplus$	XOR operation
$\Vert$	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
DA	Disconnection Acknowledgement
$RAck$	Roaming Acknowledgement
$MobResp$	Mobility Response

III. Protocols

There exist ten protocols to make the system work. These protocols define packet transfers and routes. Cryptographic primitives and the way they are used are also explained in the protocol designs.

Some protocols show similarity e.g. *Initial Authorization* and *Reuse of a Connection Card*. The only difference between these two protocols is their hash token index. *Initial Authorization* uses the very first hash token while *Reuse of a Connection* Card using the other hash tokens on the hash chain. This kind of similar protocols will be explained simultaneously.

The designed protocols are formed by the usage of some cryptographic primitives such as public key cryptosystems and hash functions forms up the designed protocol. 2048-bit RSA [3] is employed for public key encryption-decryption and signature purposes. AES-128 [4] is utilized for symmetric key cryptography and SHA-256 [4, 5] is used as a hash algorithm in the system. HMAC [5, 6] algorithm is used for challenge-response protocols.

A. *End-to-End Two-Way Protocols*

The main protocol in the system is the End-to-End Two-way protocols, which are also the most common ones in the system. The generic depiction is shown in Figure 2.

The protocols classified as End-to-End Two-way are *Initial Authorization, Reuse of a Connection Card, Disconnection, Change Alias* protocols. These protocols transmit equally sized packets from client to TTP. TTP executes the same cryptographic operations on the packet and forwards the packet to the client. In these protocols client performs an encryption over a 384-bit packet using RSA-2048 and sends it to the TTP. TTP decrypts this cipher using RSA-2048 private key then signs 256-bit data using RSA-2048 private key. TTP sends this signed data to GW through the operator. GW encrypts the response with the symmetric key between itself and the target AP and sends it to the target AP through mesh backbone.

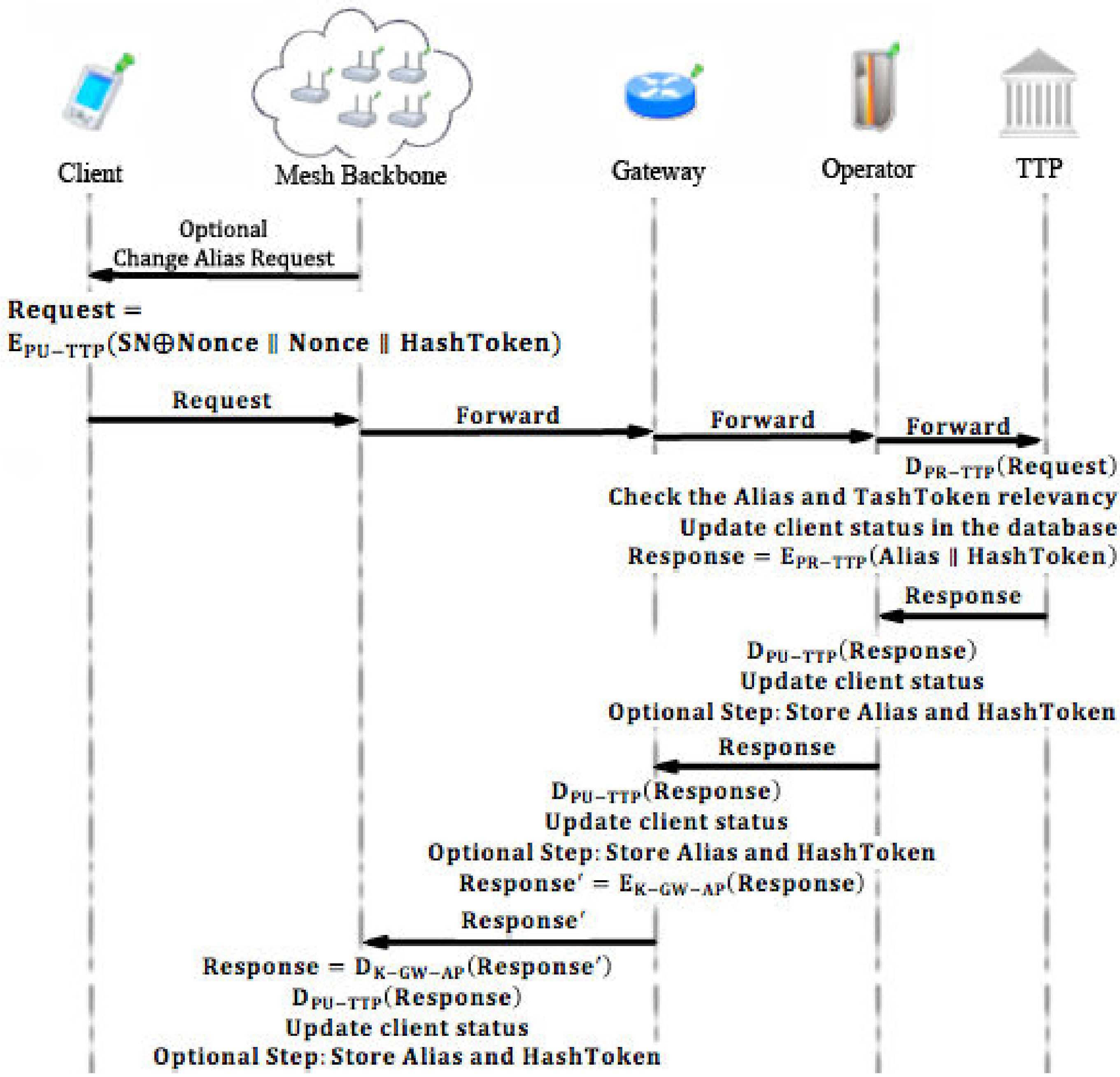


Figure 2. End-to-End Two-Way Protocol Flow

Initial Authorization is the first protocol that a client uses in the system in order to get authorized. It is used only once by a particular user. Protocol starts with client forming up a Connection request. Considering the generic depiction Figure 2 $Request = Connection\ Request$ in the case of Initial Authorization. Alias is calculated by taking the xor of Serial Number and a random nonce value as following $SN \oplus N_{CL} = Alias$. HashToken variable is H_0. When TTP receives the Connection Request (CR) it decrypts it with it's own private key and mark the client as connected in the database. In Initial Authorization protocol $Response = Connection\ Response$. When TTP receives the alias it first verifies the SN and hashes the Alias. TTP and client will calculate the alias that will be used in the system as following: $h\ (SN \oplus N_{CL}) = Alias$.

Reuse of a Connection Card protocol is used when a user does not finish the tokens in a connection card and would like to use the remaining tokens at a later time. Initial Authorization and Reuse of a Connection Card protocols only differ in their hash token index. In Initial Authorization protocol the HashToken value is H_0 whereas in Reuse of a Connection Card protocol HashToken value is H_i where $i > 0$. In Initial Authorization and Reuse of a Connection Card protocols,performing an XOR operation of SN with a random nonce forms a new Alias.

The initial time of the session for a user is stored when a user performs one of the two previously mentioned two protocols. Disconnection protocol yields the ending time of the session. In this way, the TTP learns the amount of time that the user got served. This information is used for settlement purposes. In Disconnection protocol $Request = Disconnection\ Request\ (DR)$. DR is formed as the same as a Connection Request the only difference is packet overhead, which determines the packet's aim. There are 9 protocols that are used by the client; so 4-bit packet overhead is enough for this purpose. In Disconnection protocol client does not change it's alias but uses the existing one. Therefore TTP could understand that the client with the particular alias wants to disconnect from the system.

One of the privacy preserving features of the proposed system is that access points ask every user to change their aliases from time to time. When received such a command from the access point, clients compute aliases by calculating $New\ Alias = h\ (SN \oplus Nonce')$ and send $SN \oplus Nonce'$ to the TTP for signature and hash. The overall process is called Change Alias protocol. In this protocol the optional the packet request step is executed unlike the other protocols. Every active client forms up a Change Alias Request (CAR). In the case of Change Alias protocol $Request = CAR$. When TTP receives the CAR and it decrypts the content using it's private key. Checks the last used hash token, if it is equal to the hash token that resides in the CAR then TTP signs the new Alias and the HashToken value. In this protocol TTP does not update client's status in the database because Change Alias protocol keeps a connected client connected, thus an update is not necessary.

B. *Access Point Authentication*

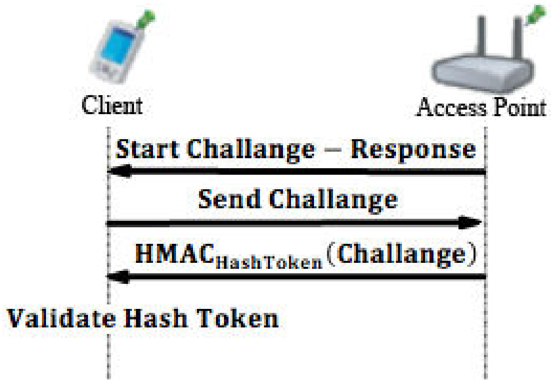


Figure 3. Access Point Authentication

Access Point Authentication, which is shown in Figure 3, takes place between a mobile client and an access point. It is a challenge-response type of protocol to authenticate the access point to the client.

Access Point Authentication starts with the serving access point by sending a request to the client. Client sends a 128-bit challenge to the access point. Access Point performs an HMAC [16] operation on this challenge using the last hash token as a key. Client performs the same operation and compares two results. If they match, the access point is verified as authenticated.

C. *Distributing Access Point Public Keys*

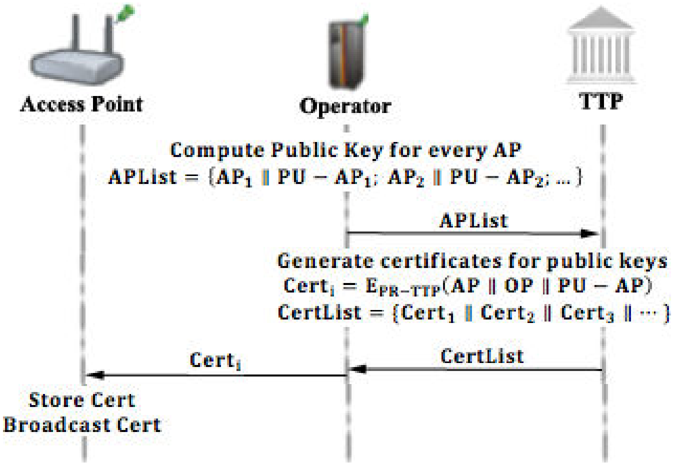


Figure 4. Distributing Access Point Public Keys

A public key distribution mechanism is placed within the system in order to achieve *Seamless Mobility* in home operator and also to support *Seamless Roaming*.

In Figure 4, a generic model for public key distribution is shown. This protocol has two parts; one is certificate generation for the access point public keys, second is distribution of the public keys. The part between operator and the TTP is offline; it runs during the set-up, before the deployment of the access points in the field.

D. *Packet Transfer*

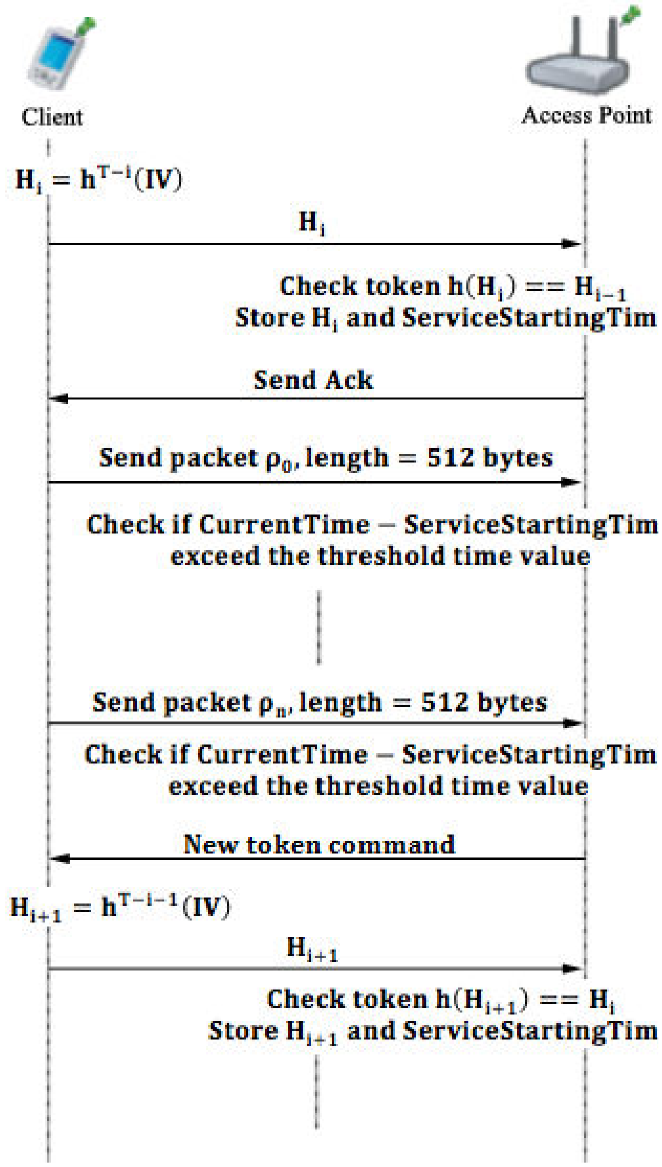


Figure 5. Packet Transfer

Packet Transfer protocol, shown in Figure 5, protocol is the simplest and the most commonly used protocol among others. It is the main service access protocol that uses tokens one by one. One token of the hash chain is sent from client to AP and the client starts to use the broadband access service. Usage is charged in time basis. Every five minutes client sends a new hash token to continue to get Internet service. When a user sends a hash token it means that she already has paid for the service and in case of disconnection the protocol is called after e.g. 2 minutes, user could not get a refund for the remaining 3 minutes.

The time measurement happens between access point and client. The access point does decrementing from 5 minutes. If client tries to get service after 5 minutes, access point sends a request to client to make her to send a new hash token.

E. *Seamless Mobility and Roaming (Payment Related)*

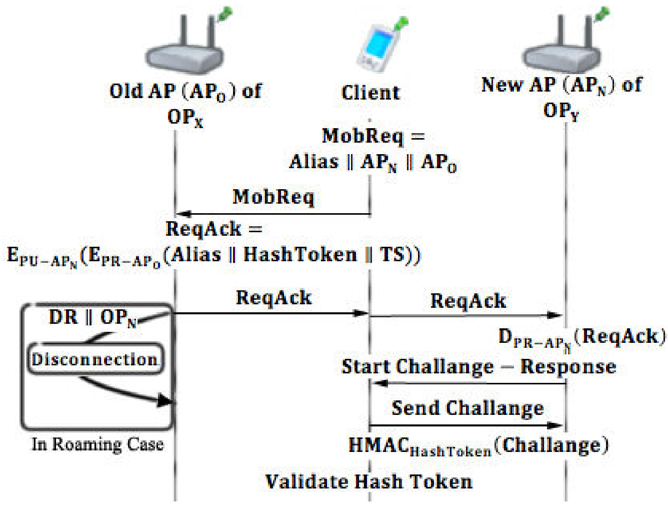


Figure 6. Seamless Mobility and Roaming

Seamless Mobility and *Roaming* protocols, shown in Figure 6, 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.*

In these two protocols client sends a 384-bit request packet to the old access point. The old access point receives this packet and performs an encryption on it using RSA-2048, than signs this cipher text using RSA-2048 private key. The old access point sends this packet to client and the client relays it to the new access point. New access point decrypts the packet using RSA-2048 private key and verifies the signature using RSA-2048 public key.

Finally the new access point and the client run a *Challenge-Response Protocol* to authenticate the new access point.

If the running protocol is *Seamless Roaming*, then receiving break-off request from the client triggers the old access point to send a disconnection request to the TTP. This part of the protocol is not implemented in the unit test because it runs in background.

F. *Update Packets*

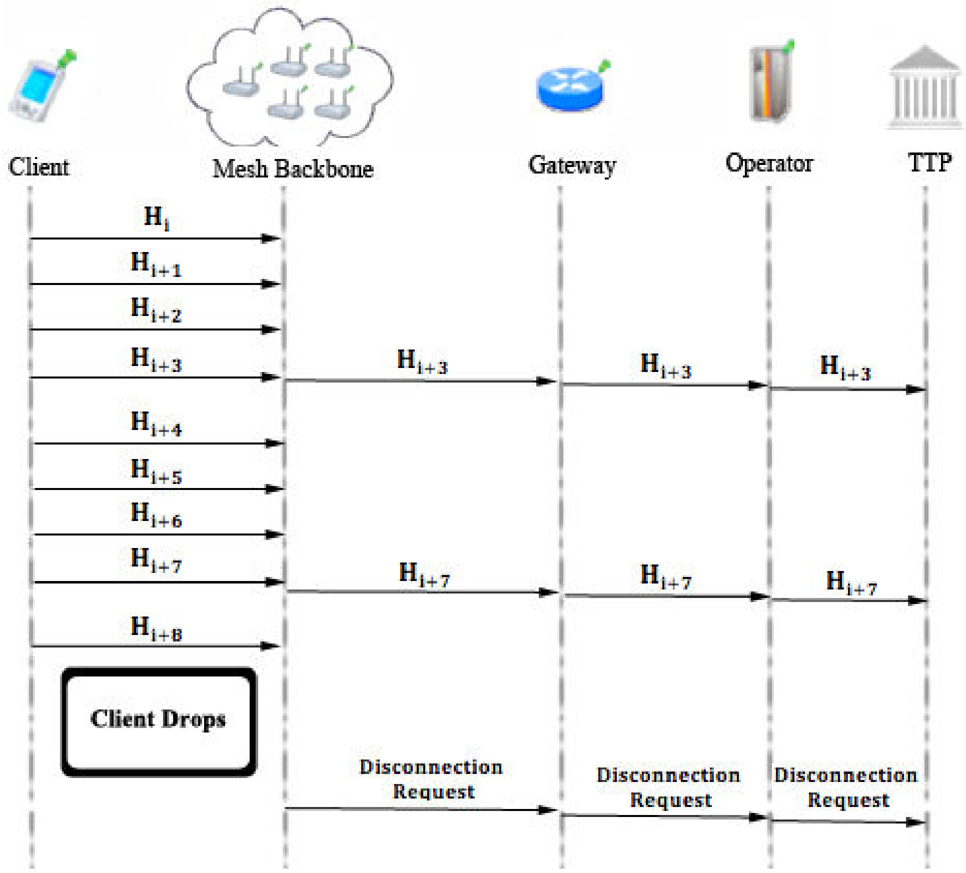


Figure 7. Update Packets

Update Packets protocol, shown in Figure 7, is used in case of an unexpected behavior in network. If a client drops out of the network, operators and TTP needs to be informed that this client is not active anymore. In order to handle this unexpected behavior, the access points periodically update operators using Update Packets protocol.

In this protocol, client sends concatenation of 128-bit alias and 128-bit hash token to the operator. Operators update TTP in case of a drop. This protocol is a one way end-to-end protocol.

IV. PAYMENT TO THE OPERATORS (SETTLEMENT)

In SSPayWMN, 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.

$$Log = OpId \parallel Connection\ Request \parallel Signed\ Connection\ Response \parallel TS$$

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.

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

TS stand for timestamp in the logs. Timestamps 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.

1. 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.
2. 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.

1. 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.
2. 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:

$$Log1 = OpID \parallel CR_{H_0} \parallel Signed\ RP_{H_0}$$
$$Log2 = OpID \parallel DR_{H_{10}} \parallel Signed\ DA_{H_{10}}$$
$$Log3 = OpID \parallel CR_{H_{21}} \parallel Signed\ RP_{H_{21}}$$
$$Log4 = OpID \parallel DR_{H_{30}} \parallel Signed\ DA_{H_{30}}$$

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.

$$Log5 = OpID \parallel CR_{H_{11}} \parallel Signed\ RP_{H_{11}}$$
$$Log6 = OpID \parallel DR_{H_{20}} \parallel Signed\ DA_{H_{20}}$$

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.

V. Simulation Environment

The network topology is hierarchical and WMN supports connections with other IEEE 802.11 protocols [14, 15], clients communicate with TTP via APs, GWs and operators in sequence. Access points are connected to gateways with 6-54 Mbps Wi-Fi connection. Some important specifications about the *AP*s are shown in Table 3. *Update Interval* determines the time value between two update packets that access point send to TTP.

TABLE III
AP Specifications

AP-Gateway Connection bit rate	6-54Mbps – Wi-Fi
AP-Gateway Distance	100m
Service Duration per token	5minutes
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 [11]. For operator servers and TTP servers, timings from [12] are used. For mobile clients, performance values from [13] are used. For AES timings the values from [21] are used, which results a 0.00004 second of delay for AES on Linksys WRT54GS. The same value is used for gateways as well. Timings of hash algorithms are taken from [22].

Platform specifications are shown in Table 4, and RSA timings are shown in Table 5.

TABLE IV
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 V
RSA-2048 Timings

	Gateway [11]	Linksys WRT54GS [11]	Server [12]	Client [13]
RSA Signing	1.3 ms	37.9 ms	8.13ms	1.8 ms

RSA Verification	47.3 ms	1529.0 ms	0.32ms	-

VI. Unit Test Results

Unit tests cover protocol behaviors 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.

A. *Results 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 8 gives the result for unit test of end-to-end two-way protocols.

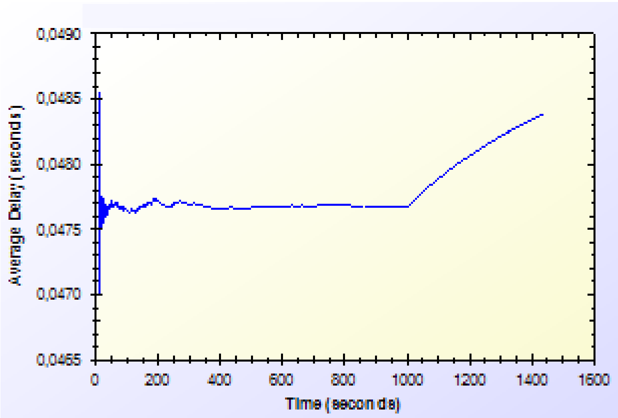


Figure 5. Unit Test Result for End-to-End Two-Way Protocols

As shown in Figure 8, 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.

B. *Results 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, shows the average delay of the protocol versus time.

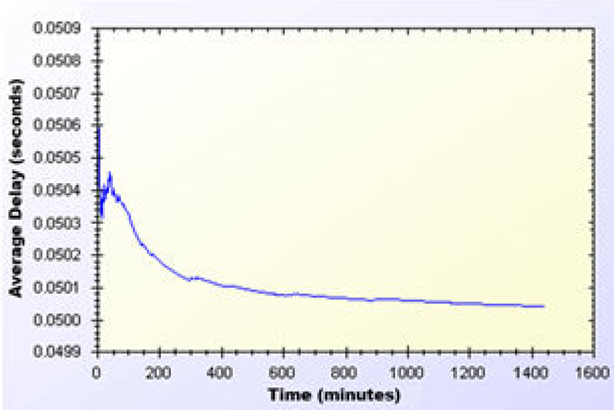


Figure 6. Unit Test Result for Access Point Authentication Protocol

As shown in Figure 9, 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.

C. *Results 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 10 presents the unit test result for Seamless Mobility and Roaming protocols.

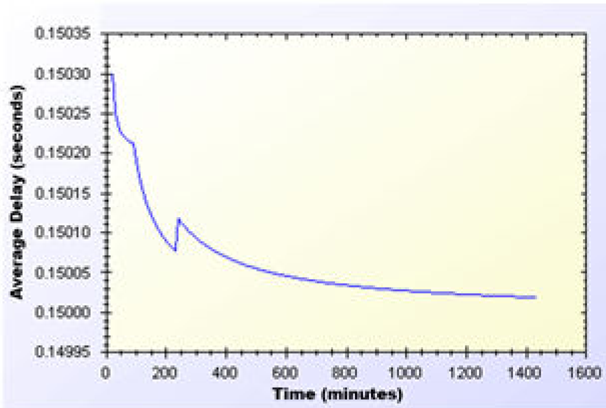


Figure 7. Unit Test Result for Seamless Mobility and Roaming Protocols

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.

D. *Results 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. Packet transfer unit test scenario is that a client sends a 512-byte packet every minute.

Figure 11 shows the unit test result for Packet Transfer protocol.

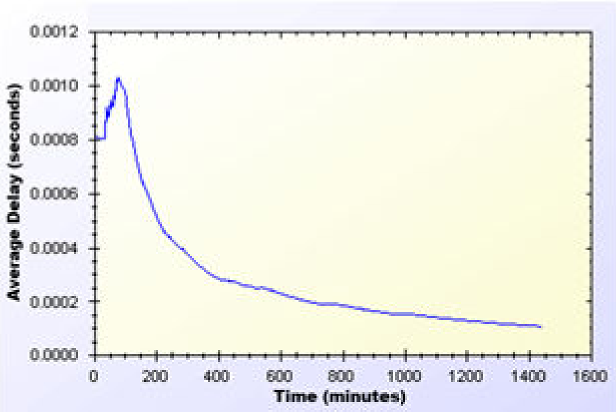


Figure 8. Unit Test Result for Packet Transfer Protocol

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.

E. *Results 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 12 shows the average delay of Update Packets protocol over time.

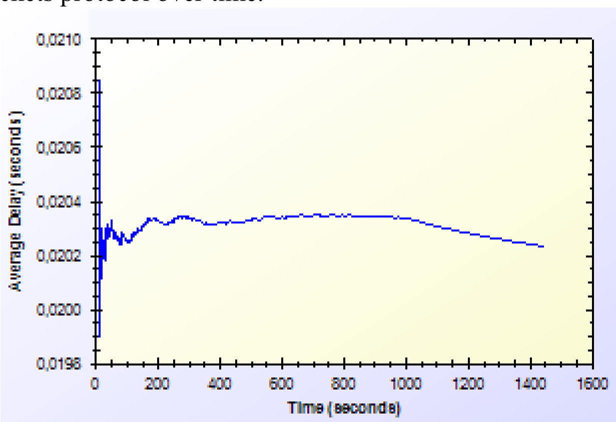


Figure 9. Unit Test Result for Update Packets Protocol

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.

VII. 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, 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 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.

A. *User Actions*

In real-life scenario simulations, network usage related actions are modeled using two-state Markov Chain as shown in Figure 13. 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.

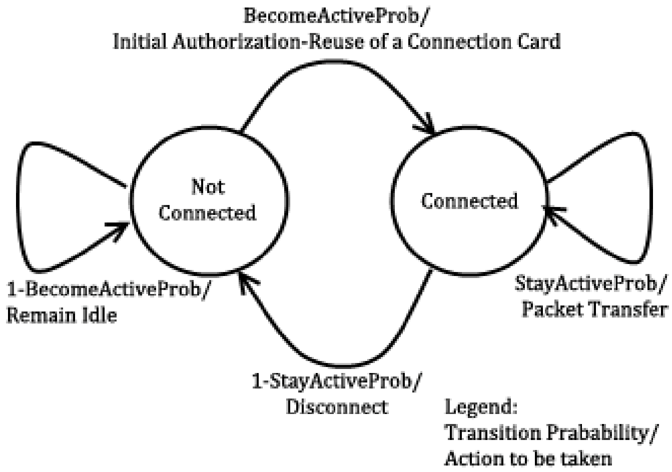


Figure 13. State Diagram of Clients

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 *CC* has been used before). In this way, the user starts consuming the hash tokens and getting Internet service. While in *Not Connected* state, the user stays in the same state with probability value of $1 - BecomeActiveProb$.

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.

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

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

where, P_{BA} denotes $BecomeActiveProb$.

B. *Client Types*

Three different user types are outlined with different networking and mobility requirements. Consideration of jobs of the clients 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 night and daytime is relatively small comparing to mid-day time. Thus, the probabilities for being active are higher for evening. Students are assumed to be more mobile than the other types of clients.
- **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. Considering mobility probability they are placed in between the other two types.
- **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.

C. *User Mobility and Timing*

Real-time scenario covers Internet usage of 300 users in a 1-km^2 metropolitan area. The simulation 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 behaviours.

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

In real-life scenario simulation 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.

Every client is 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 14. 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.

In general students travel more distance than the other client types. Random amount of distance variables are multiplied by 6 in case of student mobility. Whereas same random variable is multiplied by 3 for workers while it is multiplied by 2 for domestics. Every client moves towards the destination on a randomly assigned time. However students stay mobile more than any other types.

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.

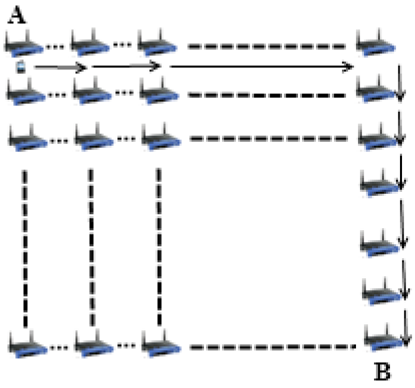


Figure 14. User Mobility

In real-life scenario simulation, there are two operators and they have same amount of access points. In current simulations, each operator has 50 access points. The client handovers or roams 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 then it means the client would perform *Seamless Mobility* protocol for handover. Otherwise, the client would run *Roaming* protocol.

VIII. Simulation Results for Real-Life Scenario

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.

A. *Overview*

Final simulations provided the results in Table 5. Charts on Figure 15 and Figure 16 are drawn exploiting the results in Table 6. 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 VI
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 Minutes	1698 Minutes	958 Minutes	16,98 Minutes
Worker	101681 Minutes	1316 Minutes	1016 Minutes	13,16 Minutes
Non-Worker	105335 Minutes	1456 Minutes	1053 Minutes	14,56 Minutes

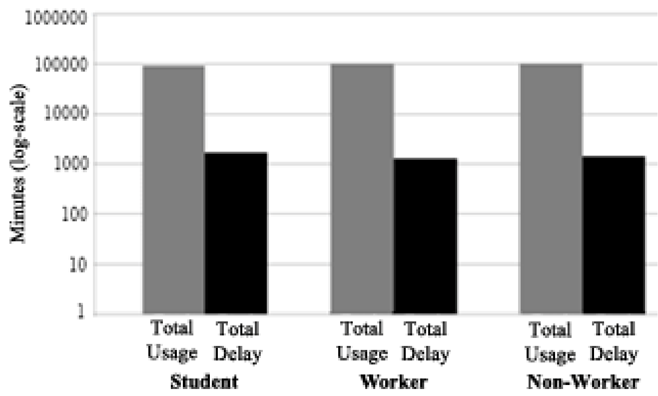


Figure 15. Total Amount of Internet Usage Times for Client Types vs. Total Delays

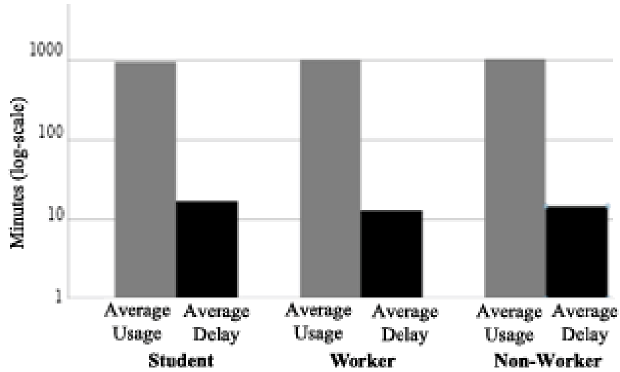


Figure 16. Average Usage Times for Client Types vs. Average Delays

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 15 and Figure 16 shows the overall results for real-life scenario simulation. Figure 15 shows comparison of minutes clients used as idle or active. Figure 16. Shows the average value for the clients of the same group.

B. *Real-Life Scenario Simulation Result for Initial Authorization*

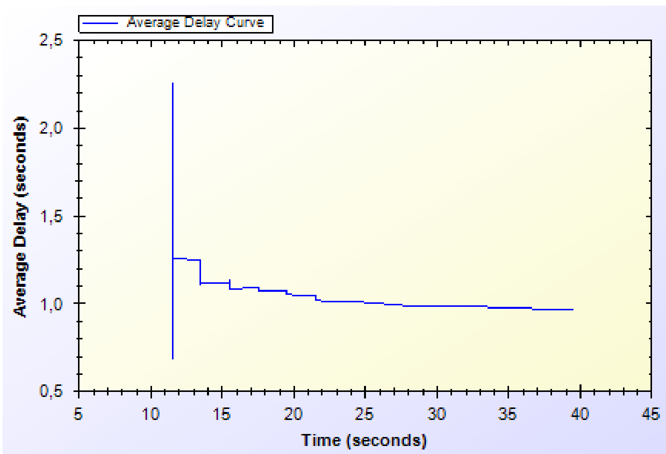


Figure 17. Result for Initial Authorization Protocol

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 17 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.

C. *Real-Life Scenario Simulation Result for Reuse of a Connection Card Protocol*

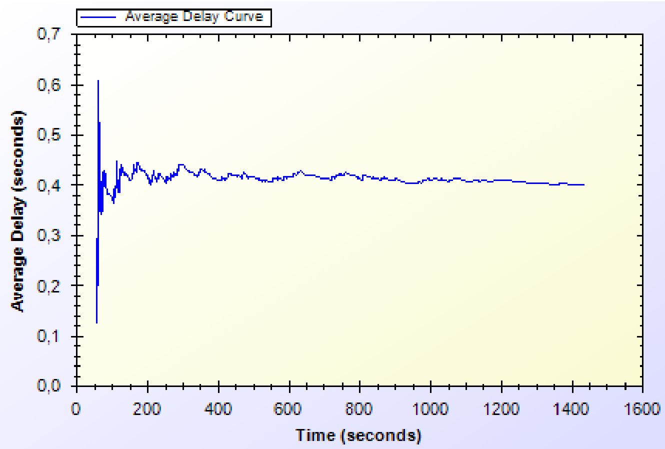


Figure 18. Result for Reuse of a Connection Card

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 18, 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.

D. *Real-Life Scenario Simulation Result for Changing Alias*

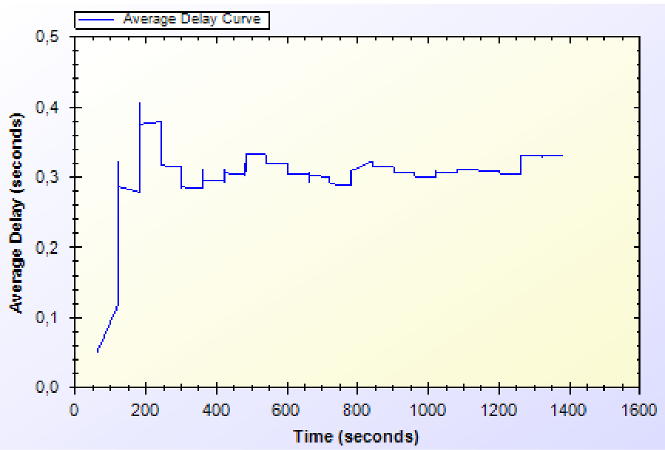


Figure 19. Result for Changing Alias Protocol

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 19, 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.

E. *Real-Life Scenario Simulation Result for Disconnection*

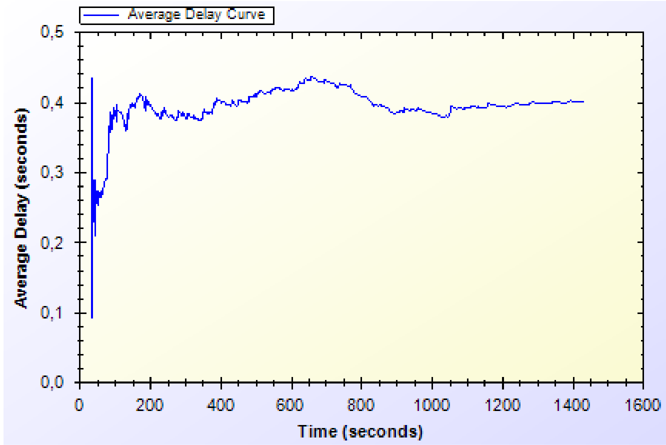


Figure 20. Result for Disconnection Protocol

Disconnection protocol first appears around 30th minute and it is used through the entire time of the simulation. Figure 20 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.

F. *Real-Life Scenario Simulation Result for Update Packets*

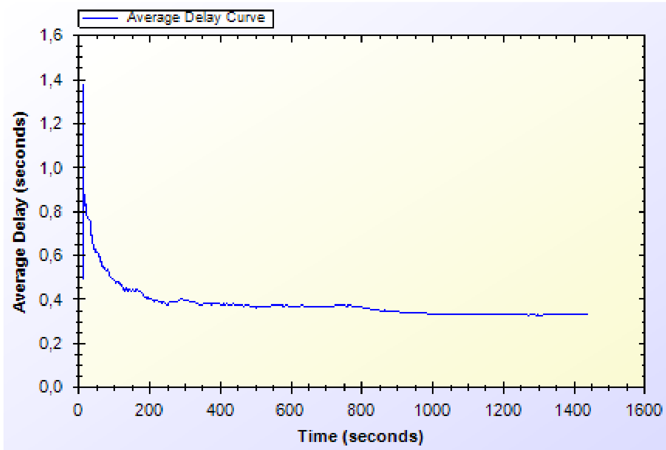


Figure 21. Result for Update Packets Protocol

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 21, 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.

G. *Real-Life Scenario Simulation Result for Seamless Mobility in Home Operator Protocol*

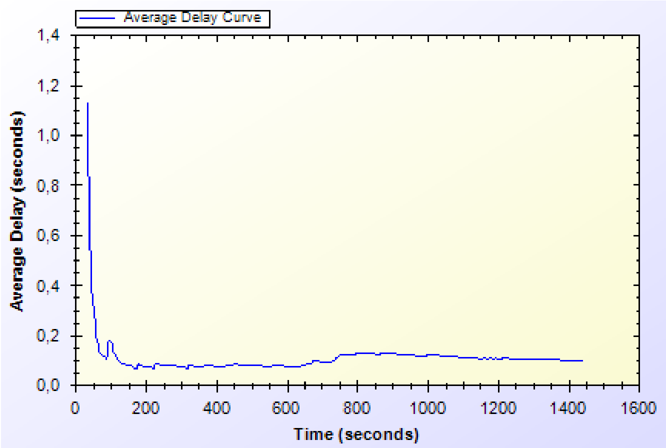


Figure 22. Result for Seamless Mobility in Home Operator Protocol

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 22, 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.

H. *Real-Life Scenario Simulation Result for Roaming Protocol*

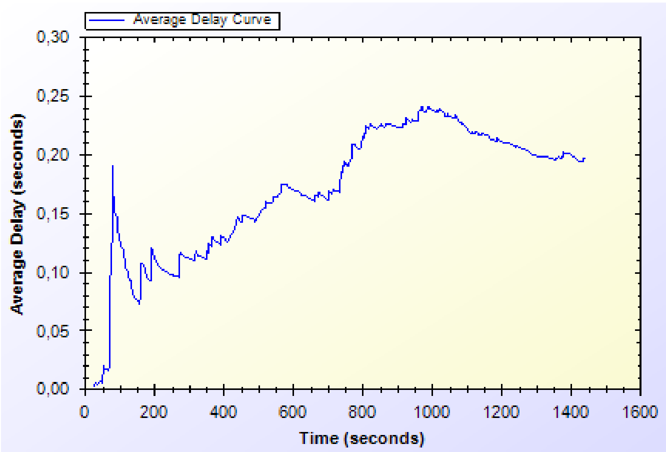


Figure 23. Result for Roaming Protocol

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 23, the results for *Roaming* protocol shows a boost until the middle of the simulation time but it decreases and achieves a balance

I. *Real-Life Scenario Simulation Result for Packet Transfer*

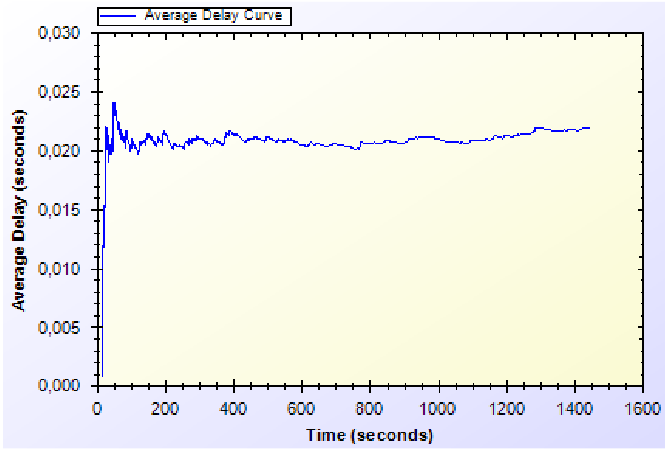


Figure 24. Result for Packet Transfer Protocol

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

Figure 24 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.

IX. Discussion

In this section the properties of SSPayWMN are discussed.

Seamless Roaming/Mobility: Clients could continue getting service without an interruption in a case of handover.

Anonymity: For legal purposes users must give their identities to Trusted Third Party (TTP) for getting connection cards. Therefore, as far as *TTP* keeps clients' identities secret, users can stay anonymous.

Mutual authentication: Challenge-Response Protocol ensures mutual authentication between AP and the client.

No ultimate trust to operators: Because of the one-way property of hash chains only the user could know the next element in the hash chain of tokens. Therefore without client giving the next element of the hash chain operator could not guess the element. The client could object to any type of over charge with cryptographic proofs.

Preventing double spending: All the connection card information is stored in the TTP's database. TTP authorizes every token; it is not possible for client to use a token for a second time. Since TTP could not get the new token with a series of hash operations.

Unlinkability: SSPAyWMN provides unlinkability 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 system designer. In real-life scenario simulations the time period was 50 minutes.

X. Conclusion

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.

Uniform probability distribution model enables us to simulate real time scenarios in simulation environment, and gets results closer to real time situations.

There exist different user types, as there are different types of clients in real life. There is also randomness inthe system, so there could be different outcomes for the same simulations. The simulations implemented to cover even the most unexpected situations.

Results are significant since the actual usage of the system is a combination of these protocols. Unit tests and real-life scenario simulation results show that the proposed system is a considerable and an effective pre-payment system.

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] Network Simulator 3 Official Web Site: http://www.nsnam.org/

[3] 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

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

[5] Trappe, W., and Washington, L. (2006) Introduction to cryptography with coding theory, *Person Education, Inc*

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

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

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

[9] 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.

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

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

[12] 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

[13] 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.

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

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

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

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

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

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

[20] 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

[21] 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.

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