Authors: Albert Levi and Can Serhat Leloğlu

Contents

1 Summary 2

2 Introduction 2

3 Protocols 3

3.1 Initial Authorization and Reuse of a Connection Card 4

3.2 Access Point Authentication 7

3.3 Packet Transfer 9

3.4 Changing Alias 11

3.5 Update Packets 15

3.6 Disconnection 17

3.7 Seamless Roaming (Payment Related) 19

3.8 Seamless Mobility in Home Operator (Payment Related) 21

4 Network Topology 23

5 Performance Results 24

5.1 Cryptographic Operations and Their Timings 25

5.2 Unit Test Result for End-to-End Two-Way Protocols 27

5.3 Unit Test Result for Access Point Authentication 28

5.4 Unit Test Result for Seamless Mobility and Roaming 29

5.5 Unit Test Result for Packet Transfer 30

5.6 Unit Test Result for Update Packets 31

6 Conclusions and Future Work 31

# Summary

Wireless Mesh Network (WMN) technology is a multi-hop, high-speed networking technology for broadband wireless access. In this project, called SSPayWMN (Secure and Seamless Payment scheme for Wireless Mesh Networks), we design a secure and privacy-preserving prepaid payment scheme for broadband access using WMN technology. The necessary requirements analyses and protocol designs were performed in the first year of the project. In the first WP of this second year, we implemented these protocols in the simulator environment and performed unit performance tests. In this document we present these unit test results for the protocols, which are acquired from Network Simulator 3 (ns-3) simulations.

# Introduction

In previous deliverables we have explained system requirements, simulator requirements, design of the system and protocols. This deliverable is mainly about the unit performances of the designed protocols.

In the first half of the second year of the project, we implemented all of the designed protocols in ns-3 simulator environment. We have performed our simulations using ns-3 (Network Simulator 3) version 3.9. We have run the simulations on 2.4 GHz Intel Core 2 Duo, 2 GB 1067 MHz DDR3, Apple Macbook OSX v10.6.8. The version of ns-2 that we are using had some bugs about mesh networks which disable the simulator to simulate packets after the 100th second of the simulation time; we fixed it to proceed. In our simulation environment we have Wi-Fi, mesh and CSMA networks connected to each other. Mesh backbone serves as a mediator between Wi-Fi and CSMA networks. Wi-Fi network uses 802.11b/g protocol whereas CSMA uses 802.3 protocols for communication. Mesh backbone uses 802.11s protocol.

The implemented protocols are group depending on their characteristics and user interaction profiles, and then unit performance simulations are performed. In unit simulations we have only one user that uses the entire system, so unit simulations give the best available results. They are proof of concept simulations that shows system is up and running. The results will have more delay when we will perform real-life scenario simulations in WP 2, but we do not expect a drastic change on the results.

Some protocols show similar behavior considering the length of packets that are sent and received, and also in cryptographic operations performed. This kind of similar protocols will differ in packet contents but they have the same delay characteristics.

The rest of this deliverable is organized as follows. In Section 3, we give an overview of the protocols that had been explained in D3 of first year in detail. Then in Section 4, the network topology that we used in the simulations is explained. Section 5 discusses the results of the unit simulations. The results are presented as charts that show average network delay vs. time. Finally, we conclude in Section 6.

# Protocols

Our protocols, which were designed in the first year of the SSPayWMN project, are as follows:

1. Initial Authorization and Reuse of a Connection Card (Reuse-CC)
2. Access Point Authentication
3. Packet Transfer
4. Changing Alias
5. Update Packets
6. Disconnection
7. Distributing Access Point Public Keys
8. Seamless Roaming
9. Seamless Mobility in Home Operator

We considered all of these protocols in the unit performance tests except one, which is Distributing Access Point Public Keys. This protocol is run before the system set up and it is not related to clients.

Table 1 shows the network entities. We have four different network entities.

Table 1. System Entities

|  |  |
| --- | --- |
| C:\Users\Public\Pictures\client.png | Mobile user (client) |
| C:\Users\Public\Pictures\ap.png | 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. |
| C:\Users\SUUSER\Documents\GitHub\worddoc\thesisImages\meshBackbone.png | Mesh backbone of the operator |
| C:\Users\Public\Pictures\gateway.png | Gateway (GW) that connects the mesh backbone to outer world and also to the operator's server |
| C:\Users\Public\Pictures\operator.png | Operator's server (OP). Keeps necessary logs and user info. |
| C:\Users\SUUSER\Documents\GitHub\worddoc\thesisImages\ttp.png | Trusted Third Party (TTP). Payment related logs are mostly to be generated by the TTP. |

## 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 3.1, 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-CC* 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 (). 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.

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.

Macintosh HD:Users:canleloglu:Desktop:worddoc:thesisImages:protocolsInDetail:newPdf:initAuthReuse.pdf

Figure .. Initial Authorization and Reuse-CC

Mobile clients introduce themselves to the operator using *Initial Authorization* protocol. in *Initial Authorization* protocol, in *Reuse of a Connection Card* protocol. TTP already knows mobile user’s serial number () and the first element, , of her hash chain. The mobile user does not want to reveal her to any adversary because that will be used all the time and it is sensitive information from security and privacy points of view. To achieve anonymity, the mobile client computes an alias and uses this value instead of . 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 that she generated.
2. (The is assumed to have credits)
3. Client sends this to .
4. Client starts to generate an anonymized subhash chain as the network processes client’s .
5. receives the connection request and relays the request through mesh backbone.
6. Gateway receives the and relays it to the operator.
7. Operator relays to TTP.
8. TTP receives the connection request () and decrypts it using its private key.
9. TTP checks alias' uniqueness within its database of users, it would make the client start over the protocol if alias is not unique.
10. It computes .
11. TTP checks and association. Store and
12. TTP computes
13. TTP generates anonymized subhash chain by taking the hash of times. The output of the last hash operation gives the first token of the anonymized subhash chain, which is .
14. TTP computes
15. TTP sends to the Operator.
16. Operator receives and verifies the signature using public key of TTP.
17. Operator sends to the gateway.
18. The Operator gets and and stores these values.
19. GW receives and verifies the signature using public key of TTP.
20. GW uses the shared secret key with and calculates
21. GW sends to the .
22. GW verifies the signature of TTP.
23. GW stores and .
24. receives and decrypts it using the shared secret key with GW.
    1. verifies the signature using public key of TTP.
    2. It calculates and and stores these values.

The wired links are secured however the medium 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.

## 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 3.2, describes the protocol briefly.

Macintosh HD:Users:canleloglu:Desktop:worddoc:thesisImages:protocolsInDetail:newPdf:accessAuth.pdf

Figure .. Access Point Authentication

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

If (a) and (b) are 3 invalid then the client sends a 128-bit challenge to the .

1. takes the HMAC of this challenge, and uses relevant anonymized subhash token as the key of HMAC.

   2. sends to the client.
2. Client also takes the HMAC of the challenge and uses the stored anonymized subhash token as the key. Then it compares the result with the one that access point sent.
   1. If it is authenticated, client starts to use access point to get Internet service.

## Packet Transfer

After mutual authentication of client and , the client starts to send data packets as shown in Figure 3.3. These packets represent the Internet usage of the client. Client could send as many packets as she wants in the period of time, which is bought by a hash token. In this protocol the usage of hash tokens and data packets are explained.

Macintosh HD:Users:canleloglu:Desktop:worddoc:thesisImages:protocolsInDetail:newPdf:packetTransfer.pdf

Figure .. Packet Transfer

1. Client starts the session with the first anonymized hash token (in this case current has value is ) of the remaining hash chain.
2. receives , and updates client’s service starting time.
   1. Checks if
   2. If true sends acknowledgement () to client and updates currently used hash value as .
3. Client sends first 512-byte data packet .
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.

The steps between (1) and (4) are repeated as long as client gets Internet service.

## Changing Alias

Anonymity property is 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. Clients generate anonymized subhash chains to break the correlation between the hash tokens that they receive service with. Clients do not fully trust to the operator and it’s access points and assume that the access points could store the hash tokens and correlate the hash tokens of different sessions.

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. 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.

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.

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.

Macintosh HD:Users:canleloglu:Desktop:worddoc:thesisImages:protocolsInDetail:newPdf:changeAlias.pdf

Figure .. Changing Alias

Changing Alias Protocol is shown in Figure 3.4 and described below.

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

When the 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.

1. Client receives "Change Alias" command.
2. Client computes a new alias by picking up a new random nonce and computes .
3. Client forms a Change Alias Request ()
4. The client sends the to .
5. Client generates a new anonymized subhash chain as the TTP processes the connection request of the client.
6. receives and relays it to the GW via mesh backbone.
7. Gateway forwards to operator.
8. Operator forwards to TTP.
9. TTP receives Change Alias Request () and decrypts it using its private key.
10. TTP checks for new alias' uniqueness and starts over the protocol if not unique.
11. TTP computes .
12. TTP computes
13. TTP generates anonymized subhash chain by taking the hash of , times. The output of the last hash operation gives the first token of the anonymized subhash chain, which is .
14. It checks SN and association and stores and .
15. It computes .
16. TTP sends to operator.
17. Operator receives .
18. Operator sends to the GW.
19. The operator computes and and stores these values.
20. GW receives .
21. The GW encrypts the and calculates
22. GW sends to the .
23. The GW calculates and , and stores these values.
24. receives and decrypts it as follows:
25. The verifies the signature using public key of TTP.

The reveals and and stores these values.

## 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.

Macintosh HD:Users:canleloglu:Desktop:worddoc:thesisImages:protocolsInDetail:newPdf:updatePackets.pdf

Figure .. Update Packets

Protocol design of Update Packets protocol is shown in Figure 3.5 and the details of the protocol are explained below.

1. After client sends the first anonymized subhash token, the access point starts to count the time passed. After units of time (value of is a system parameter, 1 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.

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.

## 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 3.6.

Macintosh HD:Users:canleloglu:Desktop:worddoc:thesisImages:protocolsInDetail:newPdf:disconnection.pdf

Figure .. Disconnection

Disconnection protocol is described below.

1. Client forms a disconnection request

* Client sends the packet to the .

1. relays to the mesh backbone, to make it reach to the GW.
2. GW receives and forwards the to the Operator.
3. Operator receives and forwards the to the TTP.
4. TTP receives the .

* TTP calculates SN and checks the relevancy between SN and .
* TTP computes the Alias, similar with the previous end-to-end two-way protocols.
* TTP computes
* TTP sends the to the Operator.

1. Operator receives .

* Operator relays to GW.
* Operator verifies the signature and marks client as disconnected.

1. GW receives .

* It relays to the mesh backbone.
* GW verifies the signature and marks client as disconnected.

eventually gets the , verifies the signature on it and disconnects the particular client, which corresponds to the it received. Ideally access points are assumed to delete all information about the past connections for the sake of freeing memory space. However if operators decide to trace user’s actions then they could do so for a limited time until the client changes it’s .

## 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 3.7, 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.

Macintosh HD:Users:canleloglu:Desktop:worddoc:thesisImages:protocolsInDetail:seqDiagram:pdf:distributeAccessPublicKeys.pdf

Figure .. 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 (); 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.

1. TTP receives the and starts to generate certificates for every access point and public key pair.

* Certificates are formed as:
* 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 together to the operator.

1. Operator receives the connection response and the certificate and relays these packets to the access point through gateway and mesh backbone.

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

## Seamless Roaming (Payment Related)

*Seamless Roaming* is run whenever the client changes the serving access point with a new one, and start to get service in a different operator’s network.

Public Key Cryptography enables us to handle seamless roaming without performing the authentication process from scratch. As it is shown in Figure 3.7, client gets a signed roaming ticket from its old access point and uses this signed ticket to maintain to get Internet service from a new access point. In parallel the old access point sends a disconnection request to it’s operator for the client.

Macintosh HD:Users:canleloglu:Desktop:worddoc:thesisImages:protocolsInDetail:newPdf:seamlessRoaming.pdf

Figure .. Seamless Roaming

*Seamless Roaming* protocol is described below. In this protocol, the client would like to switch from its old operator () to a new one (). In this setting, is the last access point that the client gets services from . , is the access point that the client would like to continue to get services in network.

1. Client sends a () to .
2. receives and forms a Roaming Acknowledgement ().

* sends to the client.
* consists of the roaming ticket that the client uses to get services from the .

1. starts the disconnection protocol for the client after sending the .

* This disconnection protocol runs in parallel with the roaming protocol. Thus it does not put an extra delay in roaming. Old operator () stores disconnection acknowledgement () 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 in order not to get confused when disconnects without a connection request reached to it.
* In this scheme, ’s signed ticket serves as a formal document, which represents the beginning of the session with .

1. Client receives and forwards it to the new operator ().

decrypts using its private key. The rest of this step is the same with (4) of *Seamless Mobility* protocol.

## Seamless Mobility in Home Operator (Payment Related)

*Seamless Mobility* is run whenever the client performs a handover between two access points of her home operator.

Every access point has its public/private key pair and ability to broadcast it’s public key, handoff in home operator could be handled in a seamless way. As it is shown in Figure 3.8, client gets a signed handover ticket from its old access point and uses this signed ticket to maintain to get the Internet service from a new access point.

Macintosh HD:Users:canleloglu:Desktop:worddoc:thesisImages:protocolsInDetail:newPdf:seamlessMobility.pdf

Figure .. Seamless Mobility in Home Operator

Seamless Mobility protocol is described below. In this setting, is the last access point that the client gets service. , is the access point that the client would like to continue to get service.

1. Client sends a Mobility Request () to .
2. receives and forms a Request Acknowledgement ().

* sends to the client.
* consists of the mobility ticket that the client uses to get services from the . It is signed by and encrypted for .

1. Client receives and forwards it to the new access point ().
2. decrypts using its private key.

* reveals the signed ticket of the . sends this signed data to it’s affiliated operator to use it for collecting funds from TTP.
* verifies the signature over this signed ticket using ’s public key. Then, it checks in order to decide whether the ticket has expired or not.
* Then, starts a challenge-response protocol with the client.

The rest of this protocol is the same as *Access Point Authentication* protocol.

# Network Topology

SSPayWMN 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 .. 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. Mesh backbone uses IEEE 802.11s. 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).

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).

# Performance Results

The simulations of SSPayWMN are conducted using ns-3. ns-3 is a widely used, popular and a free network simulator. ns-3 is a discrete event network simulator. It is mainly used in research and academia. The ns-3 project is still being developed by ns-3 users, since it is an open platform for developers. ns-3 supports 802.11s and wireless mesh networks and coding is done in C++ programming language.

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.

The network topology is hierarchical and WMN supports connections with other IEEE 802.11 protocols, 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 5.1. *Update Interval* determines the time value between two update packets that access point send to TTP.

Table .: AP Specifications

|  |  |
| --- | --- |
| AP-Gateway Connection bit rate | 6-54 Mbps – Wi-Fi |
| AP-Gateway Distance | 70 m |
| Service Duration per token | 5 minutes |
| Update Interval | 1 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 100-300-500 mobile clients.

## Cryptographic Operations and Their Timings

Platform specifications are shown in Table 5.2, and RSA-2048 timings are shown in Table 5.3. For AES timings are shown in Table 5.4. Timings of hash algorithms are presented in Table 5.5.

Table .: Platform Specifications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Gateway | Linksys WRT54GS (AP) | Server | Client |
| CPU Speed | 2.08 GHz | 200 MHz | Dual-core 64 bit 2.8 GHz | Not disclosed by Apple |
| CPU type | AMD Athlon XP 2800 | Broadcom MIPS32 | Intel Xeon | Arm Cortex-A8 |
| RAM | 512 MB | 32 MB | - | - |

Table .: RSA-2048 Timings

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Gateway | Linksys WRT54GS | Server | Client |
| RSA Private Key Operation | 47.3 ms | 1529.0 ms | 8.13 ms | 120 ms |
| RSA Public Key Operation | 1.3 ms | 37.9 ms | 0.32 ms | 3.3 ms |

Table .: AES-128 Timings

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Gateway | Linksys WRT54GS | Server | Client |
| Approximate AES-128 Timings per block | 0.001 ms | 0.01 ms | - | - |

Table .: SHA-256 Timings

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Gateway | Linksys WRT54GS | Server | Client |
| Approximate SHA-256 Timings per 256-bit block | - | 0.02 ms | 0.0002 ms | 0.008 ms |

## 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. End-to-end Two-way protocols consist of *Initial Authorization*, *Reuse-CC*, *Change Alias* and *Disconnection* protocols. Figure 5.1 presents 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.

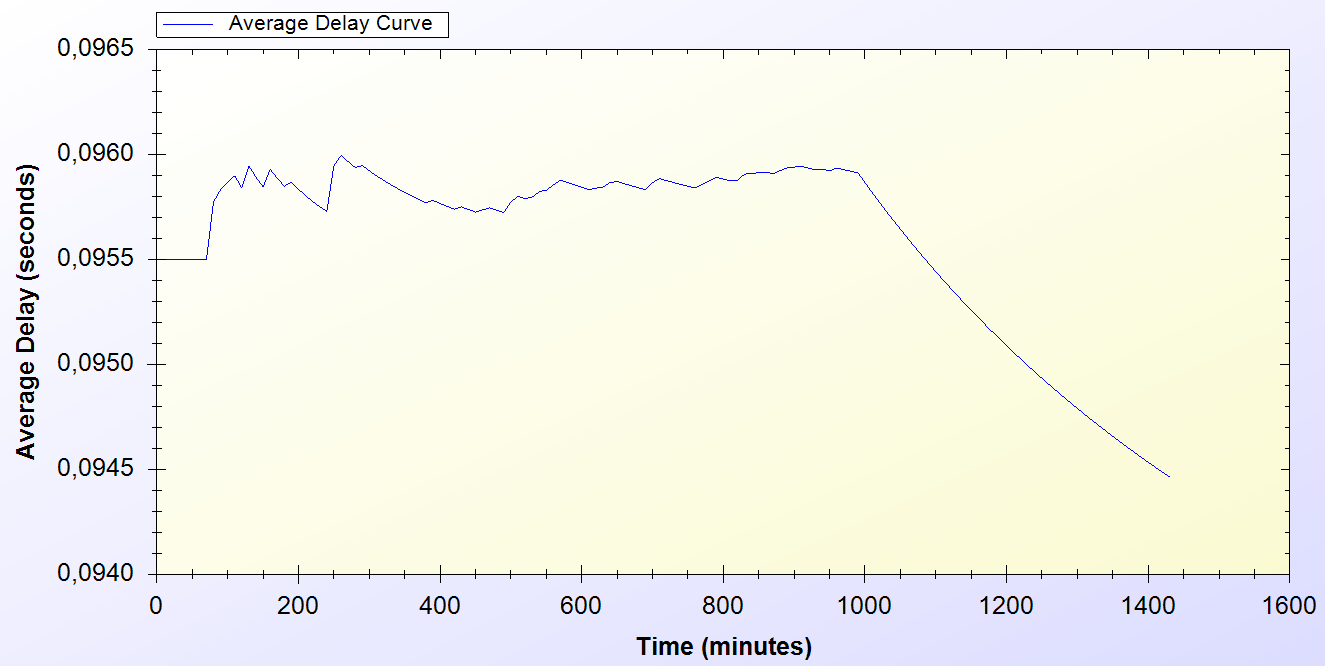


Figure .. End-to-End Two-Way Protocols Unit Test Result

There is an initial delay that shows variation around 0.1 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 and shows a significant decrease by the end of the simulation. Average delay converges to 0.0945 second.

## 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 5.2, shows the average delay of the protocol versus time.



Figure .. 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.

## 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 consist 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 5.3 presents the unit test result for *Seamless Mobility* and Roaming protocols.

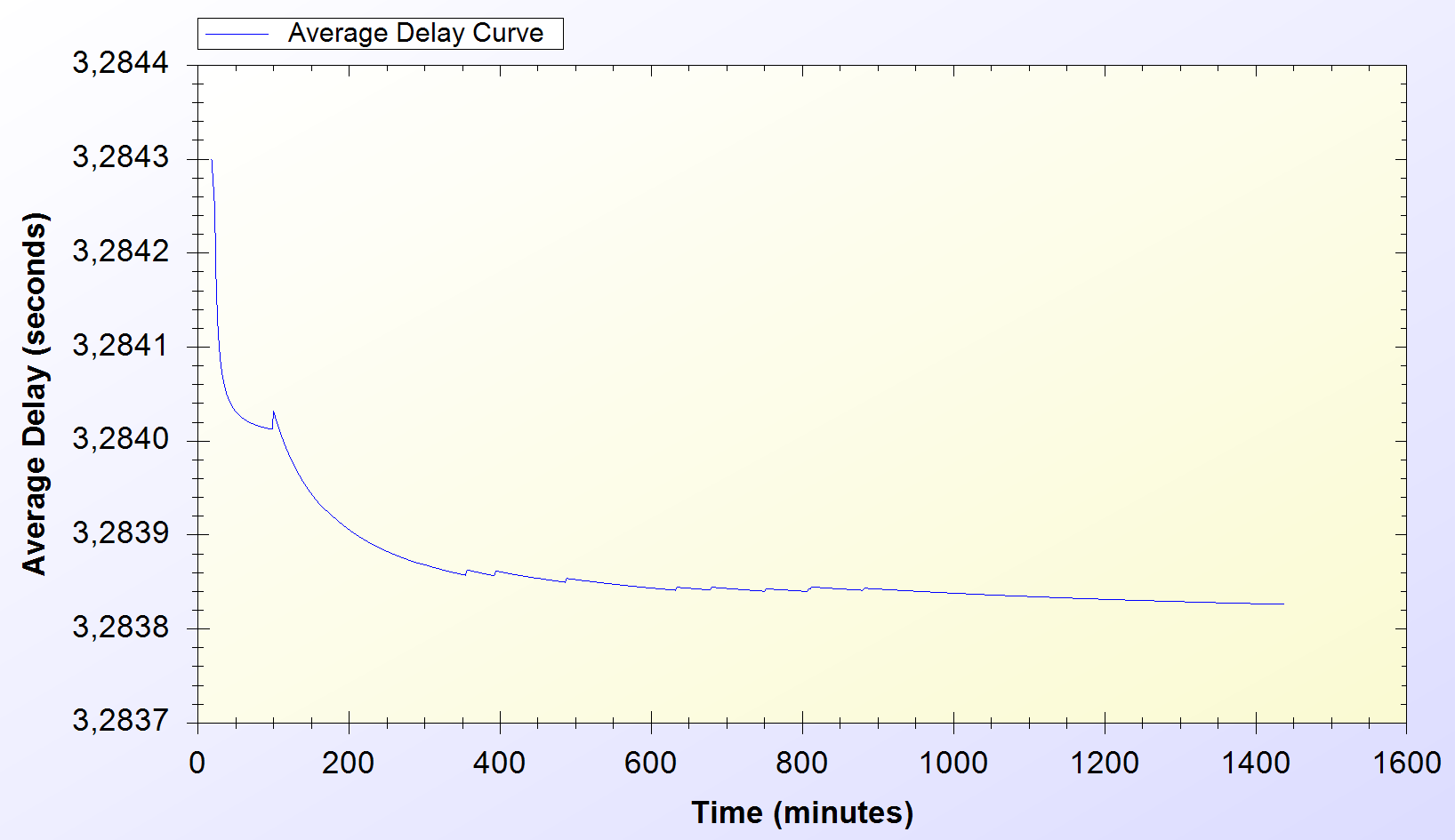


Figure .. Seamless Mobility and Roaming Protocols Unit Test Result

In unit test for these protocols, 3.28 seconds of average delay for anonymized subhash token transfer is observed. However the real network delay between service changes is approximately 0.2 second. Therefore seamless roaming and mobility is seamless to the clients. Similar to other protocols, there is a transitive period at the beginning of the simulations; however it reaches steady state in time.

## 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 10 packets of 512-byte every minute.

Figure 5.4 shows the unit test result for Packet Transfer protocol.

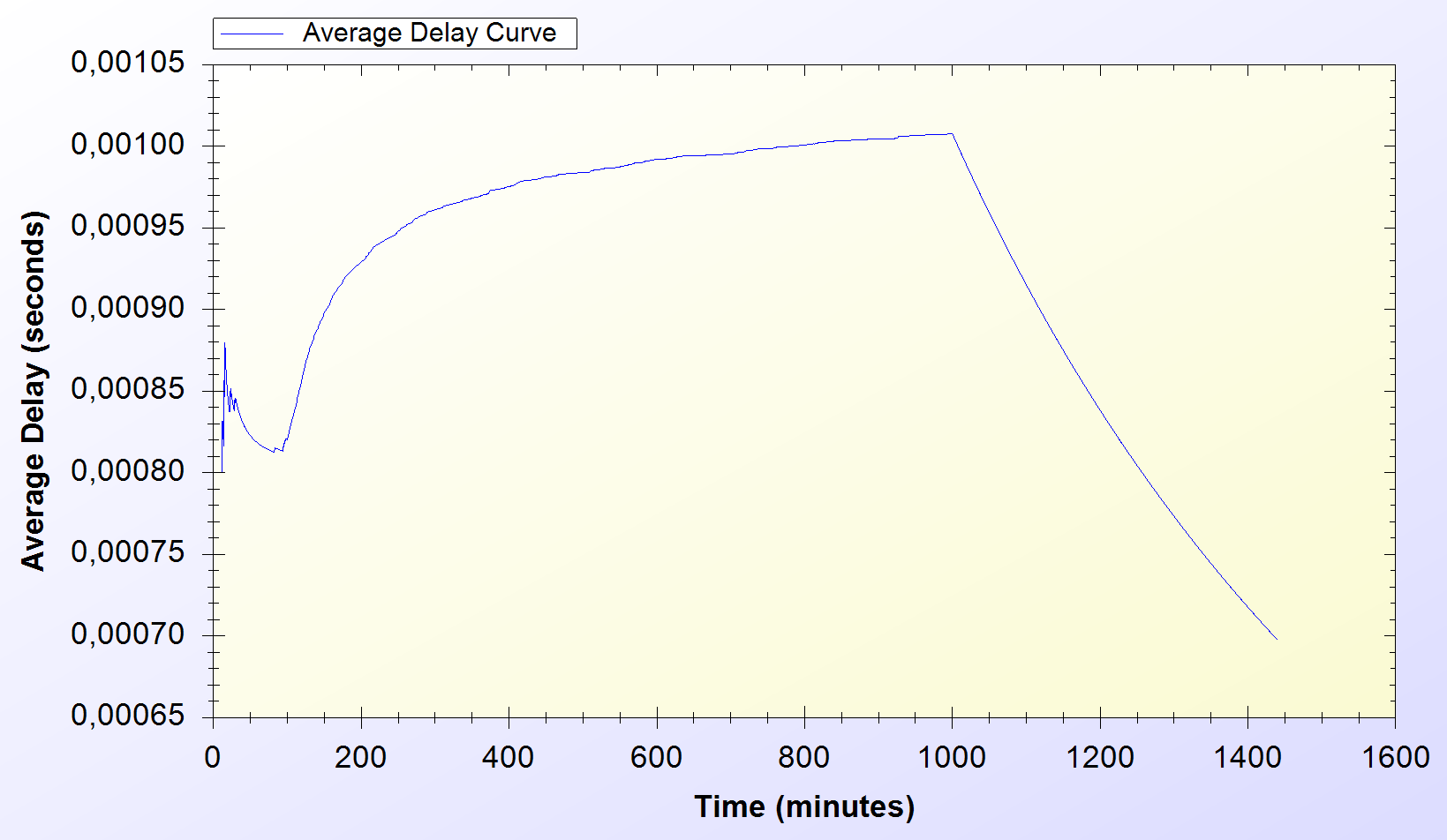


Figure .. Packet Transfer Protocol Unit Test Result

Unit test gave a higher average delay value at the early parts of the simulation but expectedly it reaches a balance through time and shows a significant decrease. At steady state, packets are received in a very short amount of time, which is around 0.0007 second.

## 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 5.5 shows the average delay of *Update Packets* protocol over time.

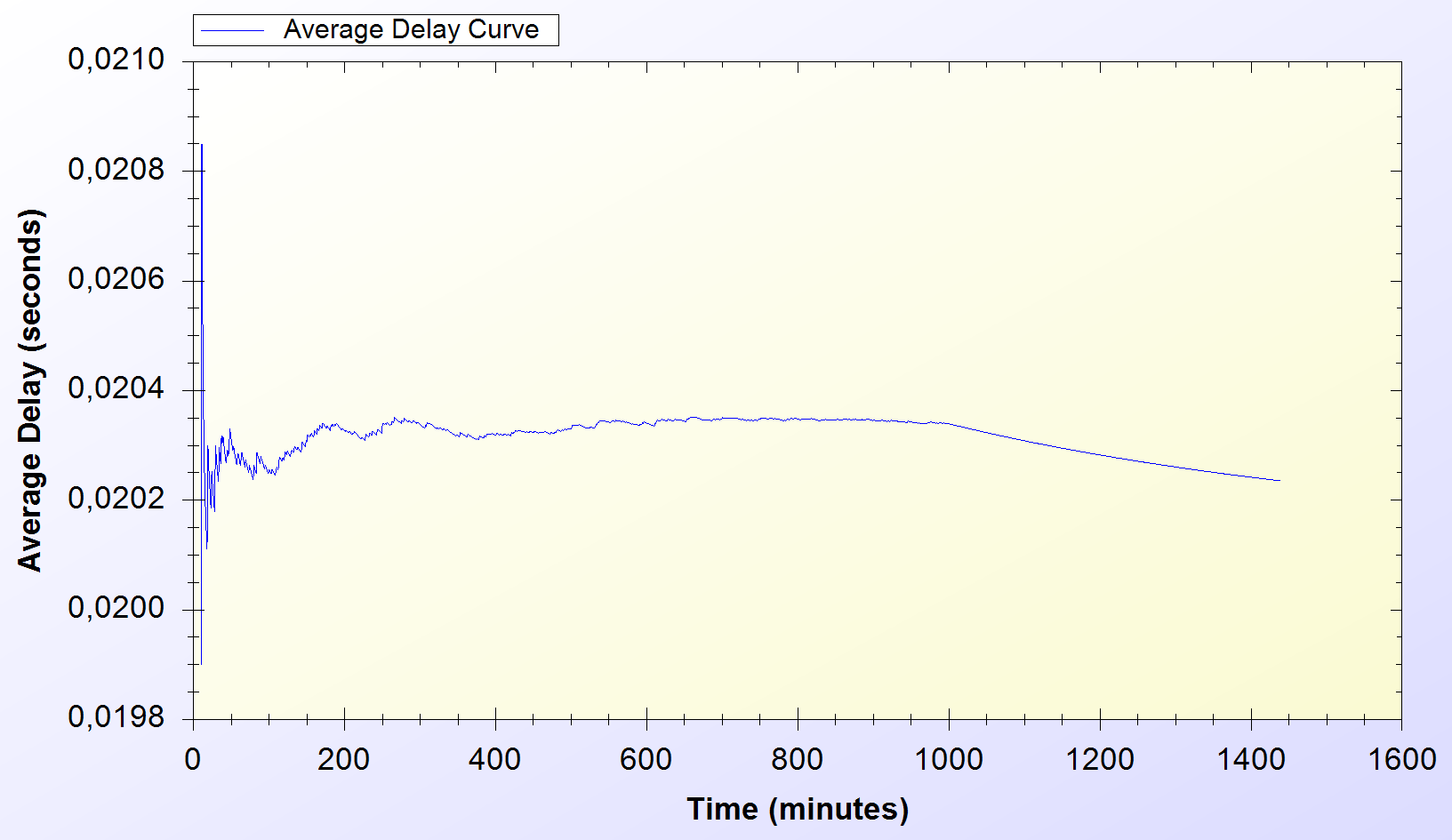


Figure .. Update Packets Protocol Unit Test Result

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.

# Conclusions and Future Work

In WP1 of the second year of our SSPayWMN project, we implemented all of the designed protocols in ns-3 simulation environment and performed unit performance test. This deliverable reports the obtained result. In these tests, we analyzed the standalone performances of the protocols under trivial usage scenarios. In other words, unit simulations set an example for how the system will behave in empty hours. In this way, we provided the first proof-of-concept implementation of SSPayWMN and showed that the designed protocols reaches steady-state and reasonable performance in time. This conclusion is very important since the actual usage of SSPayWMN is a combination of these protocols. We are very happy about our simulation results they show that our hard work paid up and gave good results to us.

As mentioned above, actual usage of SSPayWMN is combination of all of the protocols depending on the needs and system dynamics. Moreover, several users need to be considered in the simulations. Such a behavior will be analyzed in the next workpackage. In our simulations we will have different user types and their total numbers as parameters. We will also add randomness to the system, to provide different outcomes from the same simulations. The average of those simulations would cover even the most unexpected situations and this attribute of the simulations will help us to handle every possible state of the system.