Authors: Albert Levi and Can Serhat Leloğlu

Contents

1 Summary 3

2 Introduction 3

3 Protocols 4

3.1 Initial Authorization and Reuse of a Connection Card 6

3.2 Access Point Authentication 7

3.3 Packet Transfer 9

3.4 Changing Alias 11

3.5 Update Packets 13

3.6 Disconnection 14

3.7 Seamless Roaming (Payment Related) 15

3.8 Seamless Mobility in Home Operator (Payment Related) 17

4 Network Topology 18

5 Performance Results 19

5.1 Cryptographic Operations and Their Timings 20

5.2 Unit Test Result for End-to-End Two-Way Protocols 22

5.3 Unit Test Result for Access Point Authentication 23

5.4 Unit Test Result for Seamless Mobility and Roaming 24

5.5 Unit Test Result for Packet Transfer 25

5.6 Unit Test Result for Update Packets 26

6 Conclusions and Future Work 27

# 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

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

Figure 3.1. Initial Authorization and Reuse of a Connection Card

Initial Authorization and Reuse of a Connection Card are shown in Figure 3.1. Initial Authorization is the first protocol that a user uses in SSPayWMN in order to get authorized. It is used only once by a particular user. Initial Authorization is a two-way, end-to-end protocol which means client sends a packet (connection request) to TTP through all the entities of the system and receives its response back.

Client starts by making an encryption over 384-bit data packet using an RSA-2048 public key. Then the client sends this packet through mesh backbone to TTP. TTP decrypts this cipher using RSA-2048 private key and signs a 256-bit data packet using RSA-2048 private key. TTP sends this signed data to client through the mesh backbone.

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. This protocol, depicted in Figure 2, is very similar to Initial Authorization protocol. They only differ in packet content and even that content is very similar. Both protocols perform RSA encryption over a data, which is a concatenation of a 128-bit alias, 128-bit a hash token and a 128-bit nonce value.

## Access Point Authentication

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

Figure 3.2. Access Point Authentication

Access Point Authentication, which is shown in Figure 3.2, is a simple protocol which takes place between a mobile client and an access point. It is a challenge-response type of protocol to authenticate the AP to the client.

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

## Packet Transfer

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

Figure 3.3. Packet Transfer

Packet transfer, shown in Figure 3.3, protocol is the mostly used and simplest protocol among the other ones. It is the main service access protocol using tokens one by one. One token of the hash chain is sent from client to AP and the client starts to use broadband access service. Usage is charged in 1024 bytes basis. Every time client exceeds the threshold value that is predefined for a hash token, it sends another hash token to proceed.

## Changing Alias

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

Figure 3.4. Changing Alias

One of the privacy preserving features of SSPayWMN 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 a new alias and send it to the TTP for signature. Changing Alias protocol, which is depicted in Figure 3.4, after this point very similar the Initial Authorization Protocol. Client starts the protocol by encrypting 384-bit data packet using an RSA-2048 public key. Then the client sends this secured packet through mesh backbone to TTP. Then, TTP decrypts this encrypted packet using RSA-2048 private key and signs 256-bit response packet using a RSA-2048 private key. TTP sends this signed response to the client through the mesh backbone.

## Update Packets

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

Figure 3.5. Update Packets

Update Packets protocol, shown in Figure 3.5, is used in case of an unexpected behavior in network. If a client drops out of the network, SSPayWMN needs to know that this client is not active anymore. In order to handle this unexpected behavior, 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.

## Disconnection

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

Figure 3.6. Disconnection

By using Initial Authorization or Reuse of a Connection Card protocols the beginning time of the session is stored. 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.

Disconnection protocol is depicted in Figure 3.6 and operates similar to Initial Authorization protocol. Client starts this protocol by encrypting a 384-bit disconnection request message using an RSA-2048 public key. Then the client sends this packet through mesh backbone to TTP. TTP decrypts it and signs 256-bit acknowledgment using its RSA-2048 private key. TTP sends this signed ack message to the client through the mesh backbone.

## Seamless Roaming (Payment Related)

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

Figure 3.7. Seamless Roaming

Seamless Roaming protocol, shown in Figure 3.7, is run whenever client changes serving access point and this access point belongs to a different operator than that of the previous access point.

In this protocol client sends a 384-bit Roaming Request packet to old access point. Old access point receives this packet and performs an encryption on it using RSA-2048, than signs this ciphertext using RSA-2048 private key. Old access point sends this packet to client and client relays it to the new access point. New access point decrypts the package using RSA-2048 private key and verifies the signature using RSA-2048 public key.

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

After receiving break-off request from the client, old access point sends a disconnection request to the TTP. This part of the protocol is not important for the unit tests because it runs in the background.

## Seamless Mobility in Home Operator (Payment Related)

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

Figure 3.8. Seamless Mobility

Seamless Mobility protocol is very similar with Seamless Roaming protocol. The only difference between them is Seamless Mobility protocol does not run Disconnection protocol in the background.

In this protocol, which is depicted in Figure 3.8, client sends a 384-bit Request packet to old access point. Old access point receives this packet and performs an encryption on it using RSA-2048, than signs using RSA-2048 private key. Old access point sends this packet to client and 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 new access point.

# 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 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. 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 5.1: 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 5.2: 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 5.3: 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 5.4: AES-128 Timings

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

Table 5.5: 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.

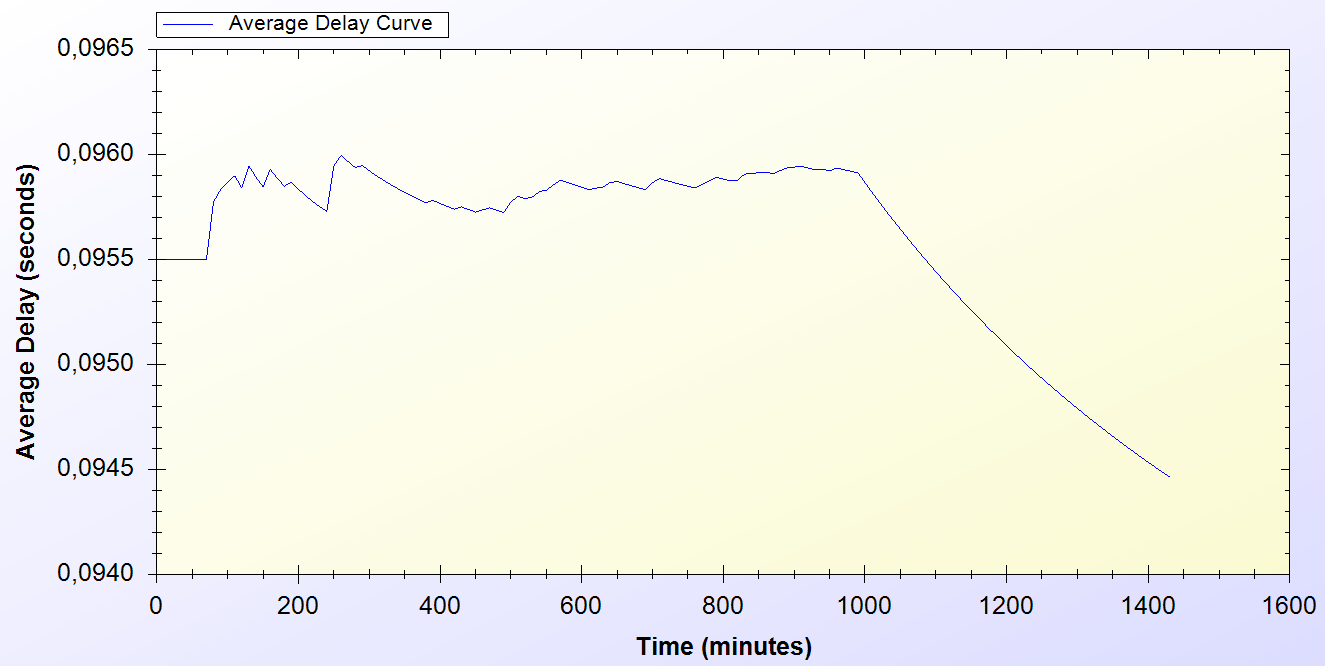


Figure 5.1. End-to-End Two-Way Protocols Unit Test Result

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 5.2. 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.

## 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.

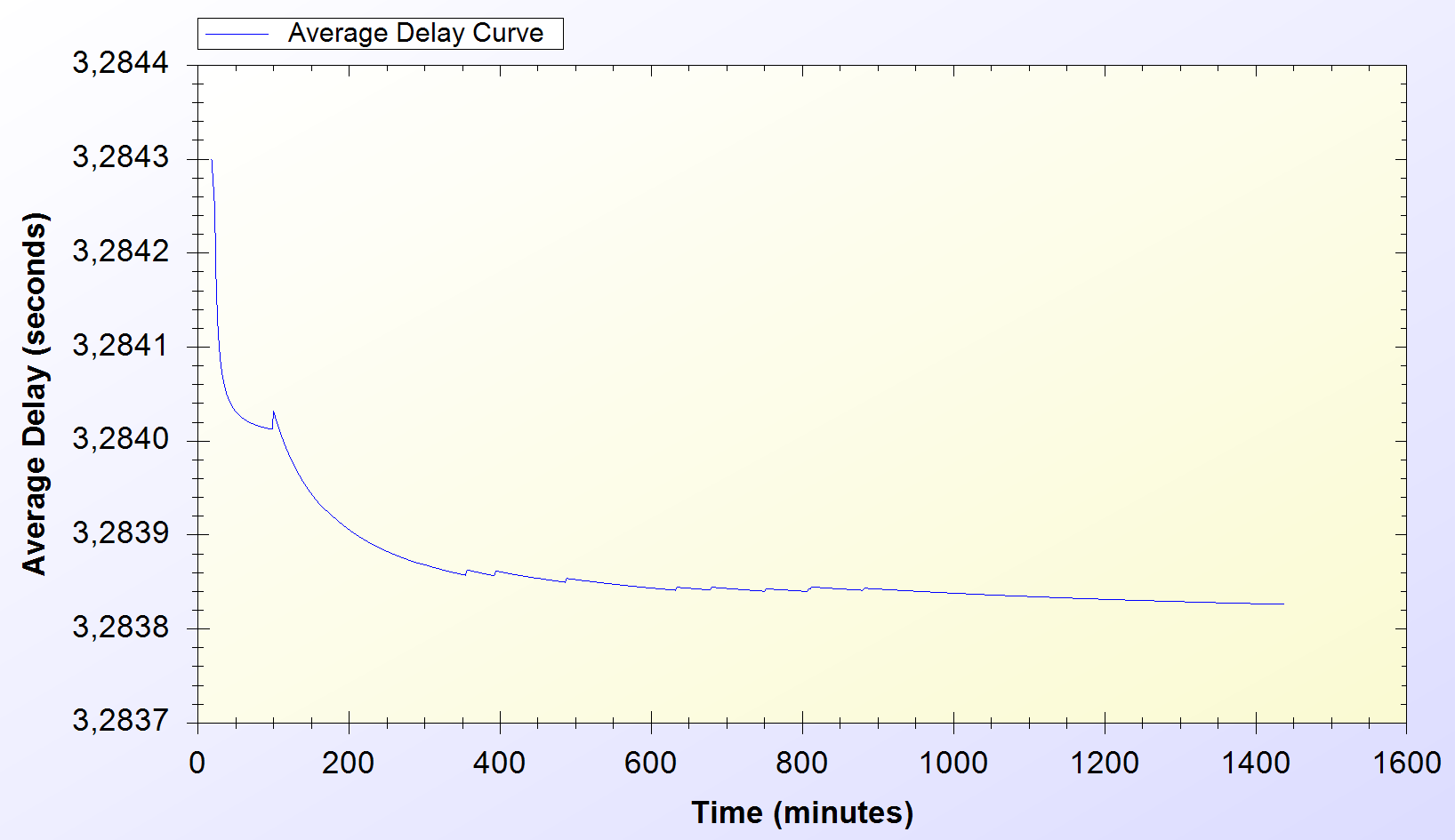


Figure 5.3. Seamless Mobility and Roaming Protocols Unit Test Result

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.

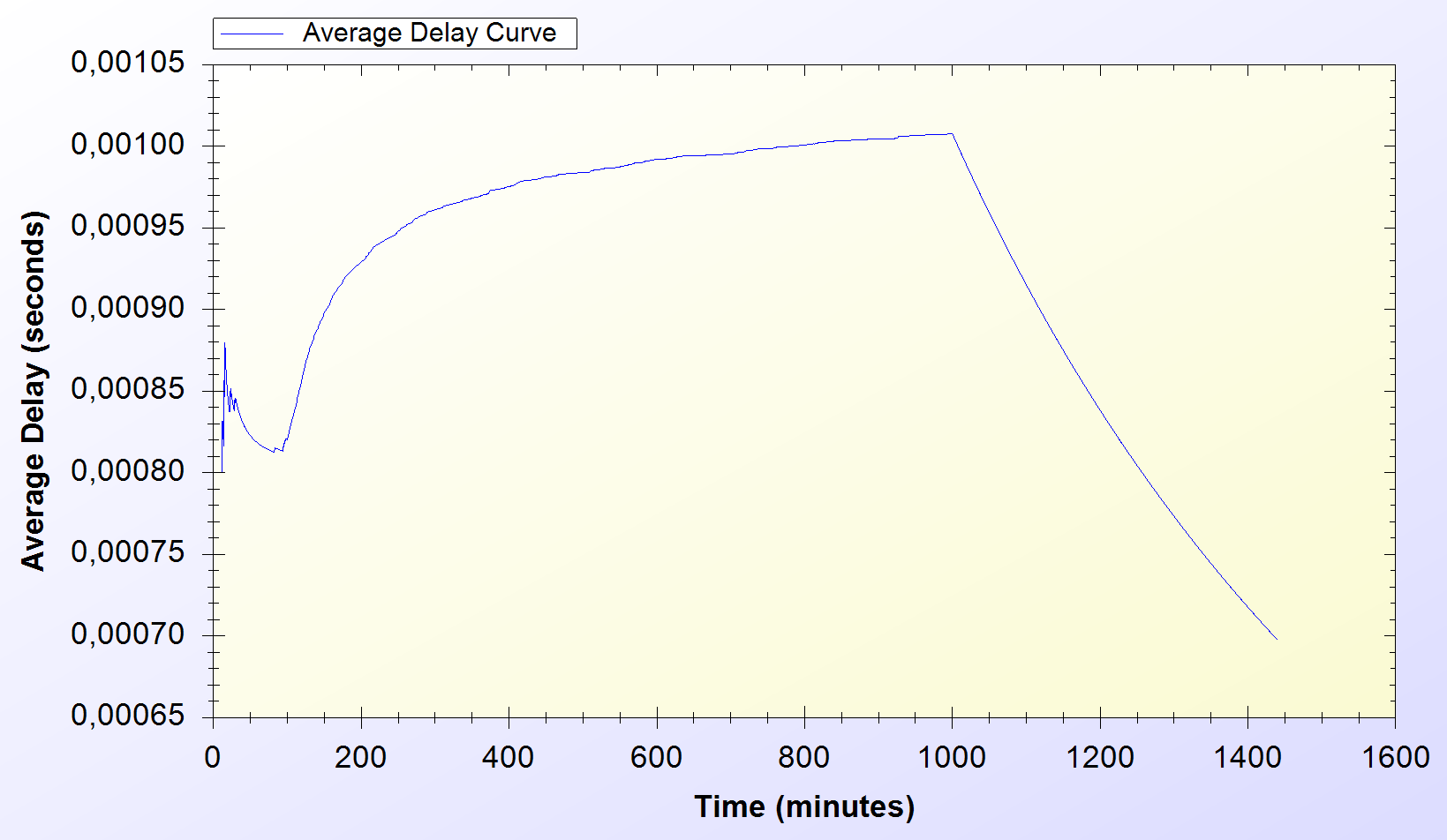


Figure 5.4. Packet Transfer Protocol Unit Test Result

Unit test gave a higher average delay value at the early parts of the simulation but expectedly it reaches a balance through time 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.

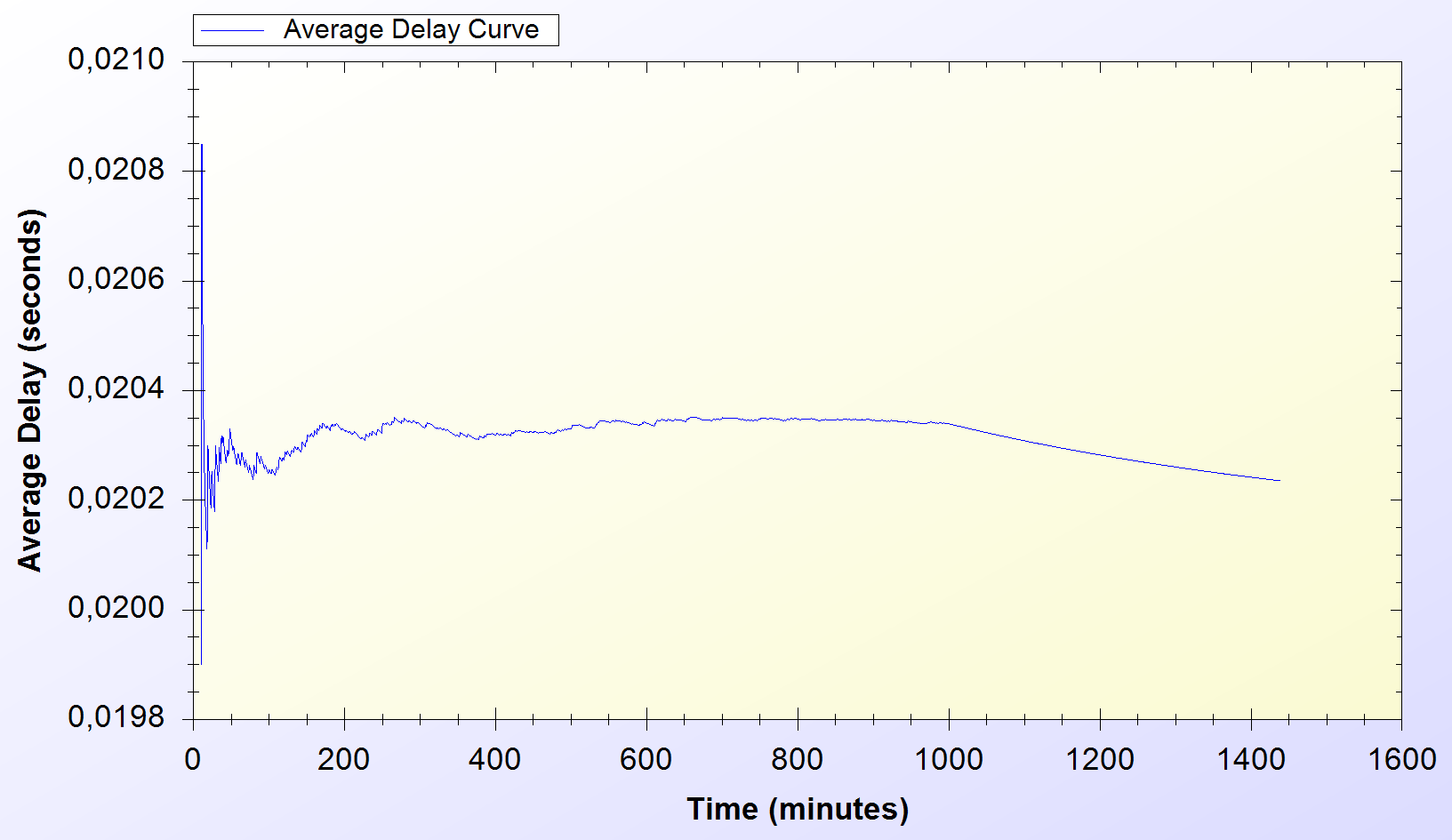


Figure 5.5. Update Packets Protocol Unit Test Result

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.

# 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.