Authors: Albert Levi and Can Serhat Leloğlu

Contents

[1 Summary 2](#_Toc337401181)

[2 Introduction 2](#_Toc337401182)

[3 Protocols 3](#_Toc337401183)

[3.1 Initial Authorization 4](#_Toc337401184)

[3.2 Reuse of a Connection Card 5](#_Toc337401185)

[3.3 Access Point Authentication 6](#_Toc337401186)

[3.4 Packet Transfer 7](#_Toc337401187)

[3.5 Changing Alias 8](#_Toc337401188)

[3.6 Update Packets 9](#_Toc337401189)

[3.7 Disconnection 10](#_Toc337401190)

[3.8 Seamless Roaming (Payment Related) 11](#_Toc337401191)

[3.9 Seamless Mobility in Home Operator (Payment Related) 12](#_Toc337401192)

[4 Network Topology 12](#_Toc337401193)

[5 Performance Results 13](#_Toc337401194)

[5.1 Access Point Authentication 13](#_Toc337401195)

[5.2 End-to-End Protocols 14](#_Toc337401196)

[5.3 Seamless Mobility and Roaming 15](#_Toc337401197)

[5.4 Packet Transfer 16](#_Toc337401198)

[5.5 Update Packets 17](#_Toc337401199)

[6 Conclusions and Future Work 18](#_Toc337401200)

# 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
2. Reuse of a Connection Card
3. Access Point Authentication
4. Packet Transfer
5. Changing Alias
6. Update Packets
7. Disconnection
8. Distributing Access Point Public Keys
9. Seamless Roaming
10. 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.

|  |  |
| --- | --- |
| D:\My Documents\albert\tt proje\accessPoint.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. |
| D:\My Documents\albert\tt proje\gateway.png | Gateway (GW) that connects the mesh backbone to outer world and also to the operator's server |
| D:\My Documents\albert\tt proje\operator.png | Operator's server (OP). Keeps necessary logs and user info. |
| D:\My Documents\albert\tt proje\trustedThirdParty.png | Trusted Third Party (TTP). Payment related logs are mostly to be generated by the TTP. |

Table 1. System Entities

## Initial Authorization



Figure 1. Initial Authorization

Initial Authorization, shown in Figure 1, 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

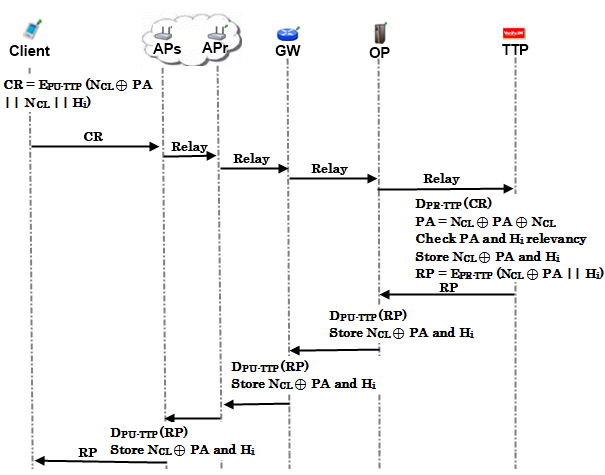


Figure 2. Reuse of a Connection Card

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



Figure 3. Access Point Authentication

Access Point Authentication, which is shown in Figure 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

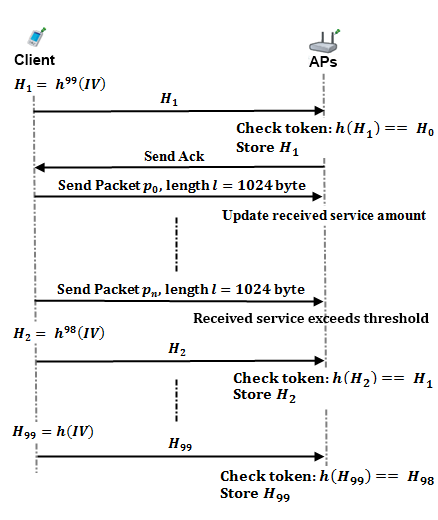


Figure 4. Packet Transfer

Packet transfer, shown in Figure 4, 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

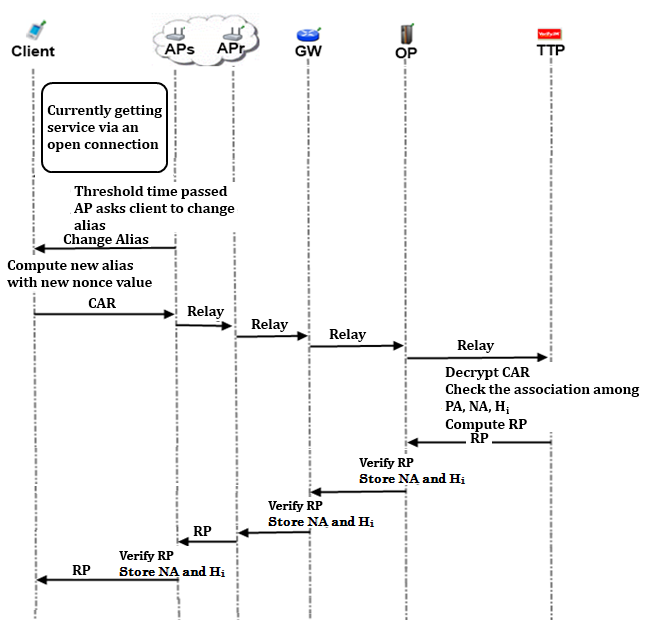


Figure 5. Changing Alias

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 5, 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

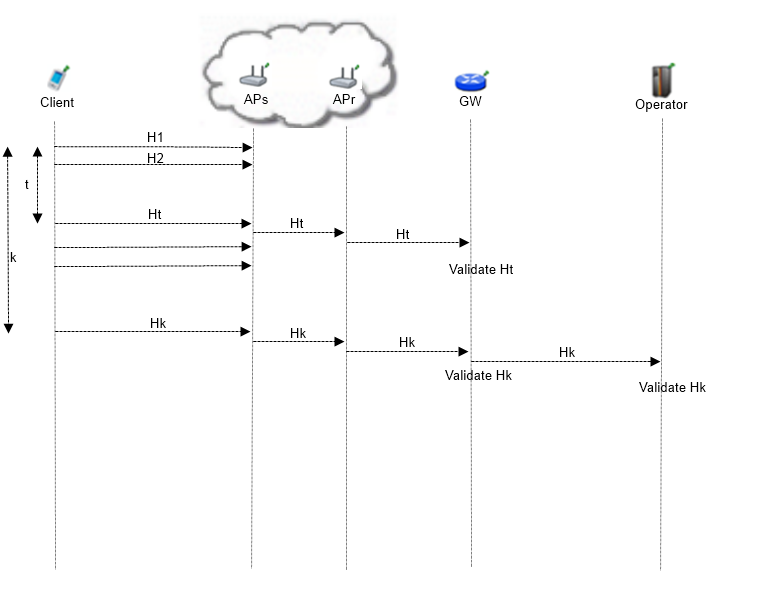


Figure 6. Update Packets

Update Packets protocol, shown in Figure 6, 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

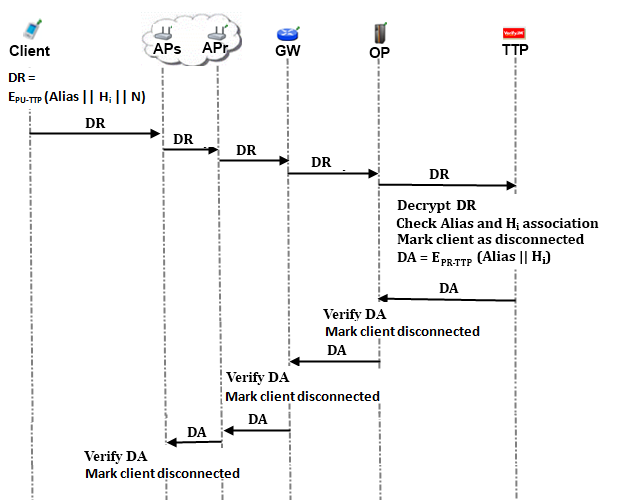


Figure 7. Disconnection

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 7 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)



Figure 8. Seamless Roaming

Seamless Roaming protocol, shown in Figure 8, 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)

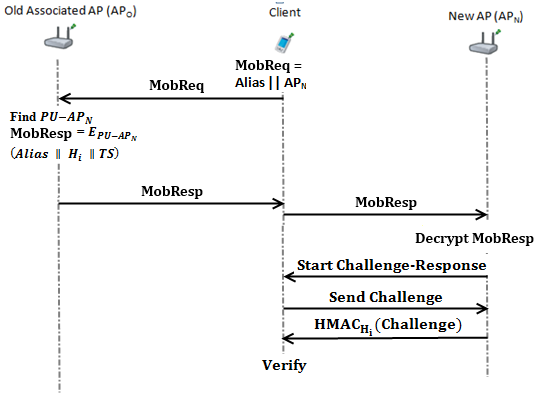


Figure 9. Seamless Mobility

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 9, 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

Figure 11 shows the network topology except the client. The unit simulation takes place when there is just one user that connects to an access point. In our unit simulation packets make 3 hops on the mesh backbone.



Figure 10. Network Topology

# Performance Results

In this section we will present our unit simulation results as charts. As mentioned earlier some protocols show similarity considering protocol properties. These protocols show same behavior in unit simulations so we have grouped them together in Section 5.2.

## Access Point Authentication

Figure 12 shows the performance of *Access Point Authentication* protocol. Access Point Authentication simulation is performed for a node, authenticating an access point once in every minute. Access point authentication process starts with a 128-bit challenge sent from the client to the access point. Then, the access point replies with a challenge. This challenge is an HMAC output over the challenge by using the last hash token as the key.

The results are shown in Figure 11. As shown in this figure, 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 considering them. That is why in ns-3 simulations and real time scenarios delays are higher at system setup.

Another important result is that we do not have unreasonable and unexpected rises or falls for delay. The average delay shows variation at the beginning of the simulation but it gets to stead state in time and shows a stable behavior. The max variation around 0.003 seconds, which is acceptable.



Figure 11. Access Point Authentication Unit Simulation

## End-to-End Protocols

Some end-to-end protocols of SSPayWMN have the same architecture. These protocols are *Initial Authorization*, *Reuse of a Connection Card*, *Disconnection* and *Change Alias* protocols. Although the content is different, these protocols are very similar to each other considering the length of packets and cryptographic operations. Figure 12 shows the performance of *Initial Authorization*, *Reuse of a Connection Card*, *Disconnection* and *Change Alias* protocols. We have performed a unit simulation for a client sending encrypted requests to operator every minute.

The results are shown in Figure 12. As shown in this figure, there is a delay that shows variation between 0.43 and 0.25. 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 in the steady-state we see average delay of 0.3 second for end-to-end communication for above mentioned protocols. Such a delay is tolerable.

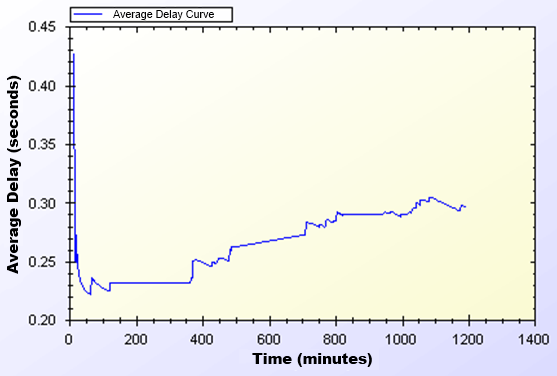


Figure 12. Initial Authorization, Reuse of a Connection Card, Disconnection, Change Alias

## Seamless Mobility and Roaming

*Seamless Mobility* and *Roaming* protocols are very crucial for our system. We need a reasonable performance for these protocols. *Seamless Mobility* and *Seamless Roaming* protocols have the same behavior since client sends and receives same length of packets. Thus, we have grouped them for unit performance evaluations. Figure 13 shows the performance of these simulations.

We have performed a unit simulation for a client changing access points in every second. The difference between seamless mobility and roaming is the serving operator of the new access point is the same with the old access point or not.

In our simulation we saw that we have 0.15 second of network delay for access point change. Similar to other protocols, there is a transitive period at the beginning of the simulations, but it comes to steady-state in time.



Figure 13. Seamless Mobility, Roaming

## Packet Transfer

Packet transfer is the mostly used protocol in our system. That’s why we need a very small amount of network delay for this protocol. Packet transfer unit simulation scenario is that a client changing access point and/or operator every minute. Figure 14 shows the performance of these protocols.

At the beginning of the simulation we have a higher network delay because of system set up. Access points talk to each other to see who is around and receive neighbor access point beacons. However, at the steady-state, we handle the requests in a very short amount of time, which is around0.0002 second.



Figure 14. Packet Transfer

## Update Packets

*Update Packets* protocol is an AP to operator protocol which means it is a one way end-to-end protocol. In this simulation access point updates the user info stored at operator. Figure 15 shows the average delay of *Update Packets* protocol over time. In the simulation scenario, APs update operator once in every second. Our simulation showed us we have a 0.08 second maximum network delay for updating operator for the client usage.

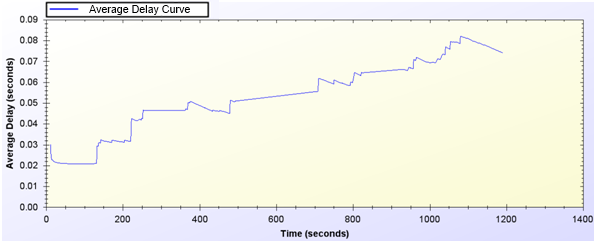


Figure 15. Update Packets

# 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.