Authors: Albert Levi and Can Serhat Leloğlu

Contents

1 Summary 2

2 Introduction 2

3 Network Topology 3

4 User Modeling and Mobility 5

4.1 User Actions 5

4.2 Client Types 6

4.3 User Mobility and Timing 7

5 Results 8

5.1.1 Real-Life Scenario Simulation Result for Initial Authorization and Reuse of a Connection Card Protocols 9

5.1.2 Real-Life Scenario Simulation Result for Change Alias 12

5.1.3 Real-Life Scenario Simulation Result for Disconnection 15

5.1.4 Real-Life Scenario Simulation Result for Update Packets 18

5.1.5 Real-Life Scenario Simulation Result for Seamless Mobility and Seamless Roaming Protocols 21

5.1.6 Real-Life Scenario Simulation Result for Packet Transfer 25

5.1.7 4.4.2. Overall Burden of the System 28

6 Conclusion 30

# 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. In the first WP of this second year, the developed protocols were tested in network simulator 3 (ns-3) as unit tests and performances were reported. In this deliverable, which includes the results obtained in the second WP of the second year of the project, we present the real life scenario results for SSPayWMN protocols. In these scenarios and corresponding tests, we model different types of users with different mobility and usage requirements changing in time of day basis, and consider various operators. As expected, the results obtained have more delay as compared to unit tests, but we achieved stable and affordable delay results.

# Introduction

In [1], we have briefly explained system and network requirements for SSPayWMN. In [2], we compared available network simulators and explained our decision for which network simulator to use. In [3], the design of the protocols are explained, and in [4] unit test results for the protocols, which are performed using network simulator 3 (ns-3), were presented. In this document we will present the results for the real-life scenario results for the designed protocols.

Unit tests [4] had been performed in order to understand the performance of our protocols under no stress. Real time scenarios, that are explained in this deliverable, show how our protocols would respond to any real time situation and how their performances are affected.

In the first half of the second year of the project we have implemented our protocols in network simulator 3 version 3.9 (ns-3). 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. We have improved ns-3 such that communication between mesh networks and other kinds of networks became possible. We also fixed a bug that was creating problems after 100th second of the mesh network simulation.

We have designed real life scenarios so that the simulation results reflect the most meaningful result. For these real life scenarios, we have grouped the clients into 3 groups. Group properties will be explained later in this deliverable. There is a probabilistic distribution for the client behavior based on the predefined group parameters and the simulated time of day so that we can model daily routines. Our clients are mobile so that their access points and/or operators change over time as well.

The rest of this deliverable is organized as follows. In Section 3, we give the network topology and design used in simulations. In Section 4, User modeling and mobility issues are explained. We give the simulation results in Section 5. Finally, Section 6 discusses and summarizes the conclusions reached by these analyses.

# Network Topology

SSPayWMN simulations are aimed to be close to real life situations. That understanding would ease the process, implementing SSPayWMN in real life. ns-3 is a successful network simulator but in order to get accurate results, a realistic network topology and user modeling need to be used. The former one, network topology, is going to be explained in this section.

In SSPayWMN, we have for different network entities in the construction of the network topology (other than the end users). These are shown in Table 1 (end users will be explained in Section 4).

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. |

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 3.1. Network Topology

Figure 3.1 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).

Real-life scenario consists of 100, 300 and 500 mobile clients trying to get network service in a 1km2 metropolitan area. This scenario simulates an ordinary day with 24 hours. In these simulations clients have mobility patterns as mentioned before. Client’s network usage frequency is affected by their socio-economical status. In following sections we will give latency values for each protocol. Finally, we will explain the overall burden on the system caused by the system’s protocol.

# User Modeling and Mobility

In SSPayWMN, we intend to serve a variety of users (a.k.a. network clients). Network clients differ in their network usage frequency with respect to time of day, their mobility patterns and roaming behavior.

In our simulations, we define user types and model their corresponding behaviors. We define certain kinds of actions, such as authorization (initial or reuse of a connection card), disconnection, packet transfer (network usage), payment related roaming and payment related AP handover. All of these actions are triggered as a result of a random event. Connection and network usage related actions are triggered according a two-state Markov Chain model, which will be described in Section 4.1. Roaming and handoff related actions are triggered by user mobility, which will be described in Section 4.3.

Three different user types are defined with different networking and mobility requirements. The differentiation among these types is provided by considering whether they are working, studying or domestic. These user types will be discussed in detail in Section 4.2.

## User Actions

In our simulations, network usage related actions are modeled using two-state Markov Chain as shown in Figure 3. 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 3. State Diagram of Clients

The initial state is *Not Connected*. In this state, a user switches to *Connected* state with the probability value of . This state transition triggers Initial Authorization (if the connection card is used for the first time) or Reuse of a Connection Card protocol (if the connection has been used before). In this way, this user starts using the network and getting the service. While in Not Connected state, a user stays in the same state with probability value of .

While in *Connected* state, the user stays connected (i.e. stay in the same state) with probability . This 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. This transition disconnects the user via Disconnection protocol.

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

Eq. 1

where, denotes .

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

Eq. 2

where, denotes .

## Client Types

Three different user types are defined with different networking and mobility requirements. The differentiation among these types is provided by considering whether they are working, studying or domestic.

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

User types are described as follows:

* **Students:** This kind of clients uses network services mostly in the evening when they return back from school. Their possibility to use network services during morning and evening is relatively small comparing to mid-day time. Thus, the probabilities for being active are higher for evening. Students are assumed to be mobile at the beginning and end of the *daytime* slot since they go to their school. Until the end of the *evening* slot, students would more likely to get service in their homes in an immobile way.
* **Employees:** This kind of clients has routine lives. They are immobile and not so active during nights and evenings. However, during the daytime, they are very active and use network services at their work places. Moreover, they are mobile as they commute to/from work from/to home at the beginning and end of the working times.
* **Domestics:** This type of users does not work outside and spend their time at home. usually they get Internet service in an immobile way. These users are highly active at all times.

The parameters of and are determined based on the abovementioned discussion about the client type characteristics and the time slots. These values are given below. The triplets specify the probability values for night, daytime and evening, respectively.

These values also determine the average connection duration and idle time by using Eqs. 1 and 2. For example, a domestic client remains idle during daytime for minutes between connections. Once connected, average connection time for this category is minutes.

## User Mobility and Timing

Real-time scenarios cover Internet usage of 100, 300 and 500 users in a 1-km2 metropolitan area. The simulation time begins at 06:00 a.m. and lasts for 24 hours. Simulation time is divided into 3 parts considering morning, daytime and evening. Every part of the day has different statistical values for client behaviors.

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

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

Clients are assigned a random target access point. Every one of 100 access points has 5 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 4.1. 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 the closest to the client’s current location



Figure 4.1. User Movement from A to B

In real-life scenario simulations, there are two operators and they have same amount of access points. In current simulations, each operator has 50 access points. The client executes handover or roaming if there is an active connection during movement between access points. In such a case, depending on the new access point’s affiliated operator, user’s movement triggers either Seamless Mobility or Roaming protocols. If new access point’s affiliated operator is same as the one that client currently uses, and then it means the client would perform Seamless Mobility protocol for handover. Otherwise, the client would run Seamless Roaming protocol.

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

# Results

Results for unit test simulations are available in [4], but the more important results are real-life scenario simulation results. Please note that there is randomness in the system. The actions of the clients are based on random numbers, but of course we define the chances they have to act in a particular way considering client type and time.

In this section, we will briefly explain each protocol’s run time performance. We have simulations for 100, 300 and 500 clients, 100 access points, 32 gateways, 2 operator servers and a server of the TTP. Every protocol in real-life simulations are executed simultaneously; therefore, real-life scenario simulation results have bigger latency values than unit tests.

In real-life simulations client models and mobility schemes are considered. The client roles and systematic mobility gave us realistic simulation results. In the beginning, every access point starts with 3 initial clients but then these clients move randomly in the metropolitan area and the initial setting do not remain as it was in the beginning.

### Real-Life Scenario Simulation Result for Initial Authorization and Reuse of a Connection Card Protocols

*Initial Authorization and Reuse of a Connection Card (Reuse-CC)* protocols are used to start service from the system. Initial Authorization protocol only occurs at the beginning of the simulations; however, *Reuse-CC* protocol takes place frequently in the system.

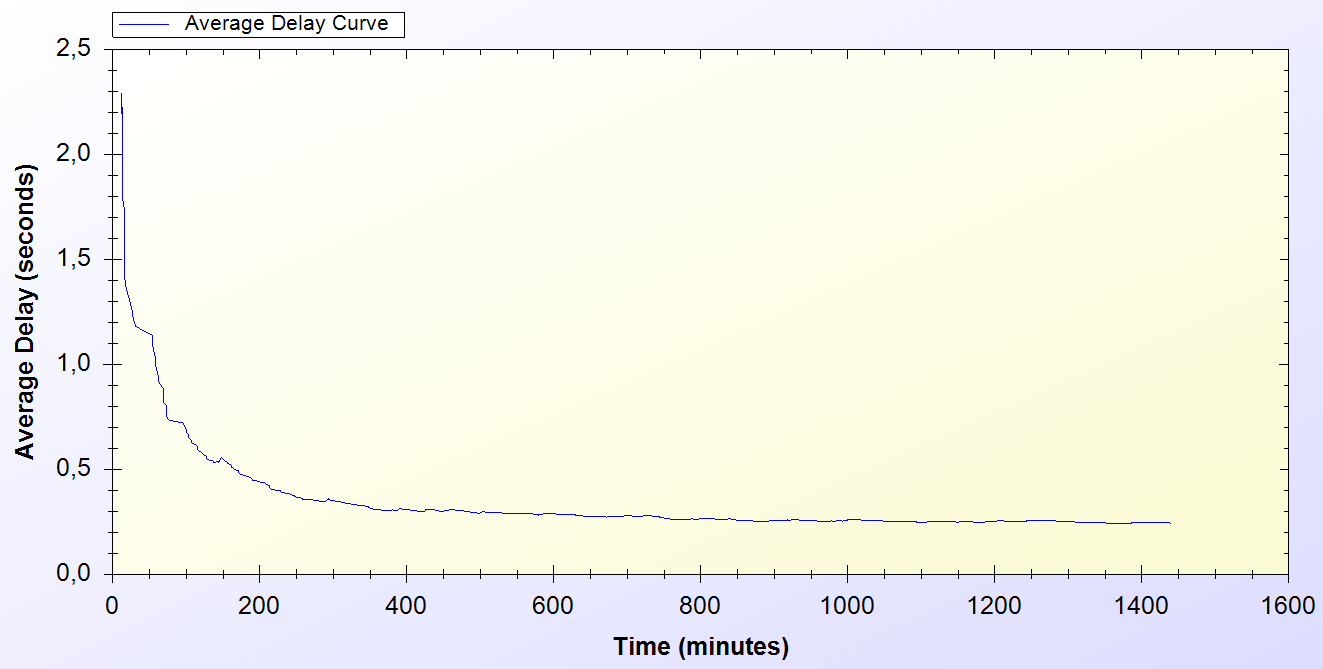


Figure 5.1. Real-Life Simulation Result for Initial Authorization and Reuse-CC Protocols (100 Clients)

Figure 5.1 shows Initial Authorization and Reuse-CC protocol performances in the real-life scenario simulation with 100 clients. The average delay for these protocols starts from 2.3 second. After some time, the protocol achieves a steady state performance of 0.3 second.

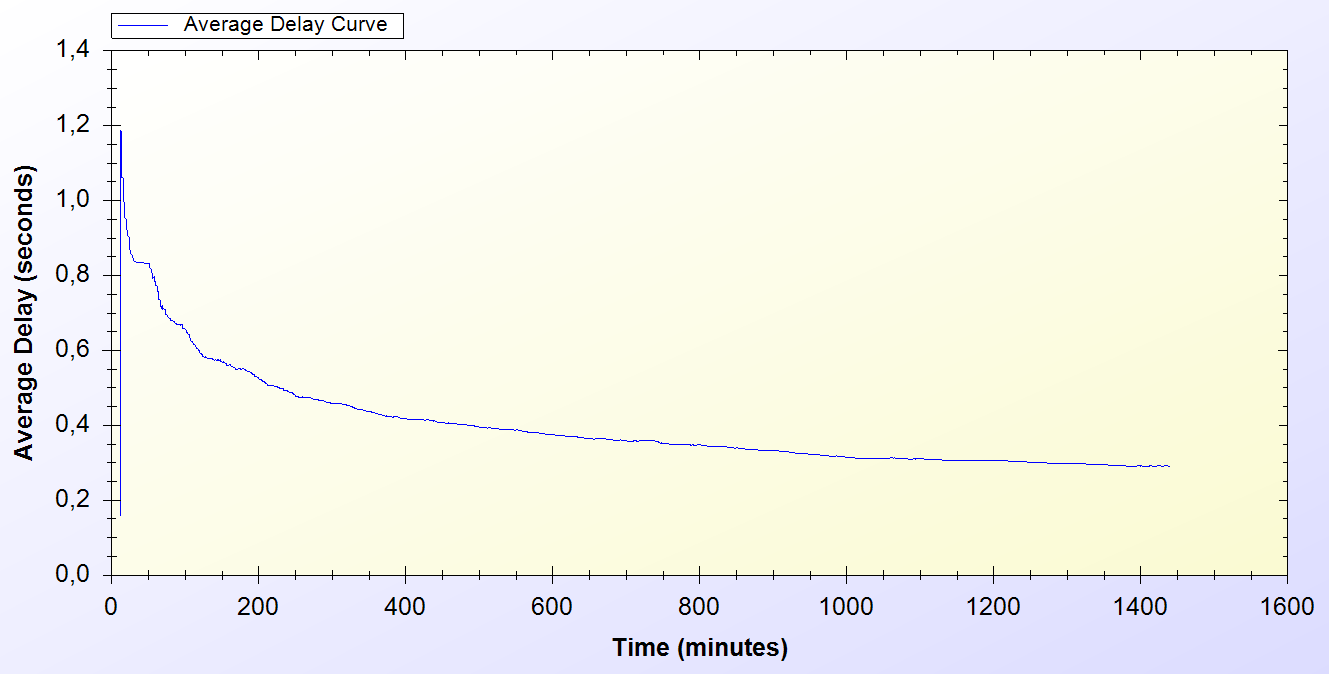


Figure 5.2. Real-Life Simulation Result for Initial Authorization and Reuse-CC Protocols (300 Clients)

Figure 5.2, shows the same kind of simulation results as the one previous. However, this time there are 300 clients in the system. With this setting, Initial Authorization and Reuse-CC protocols show variation on delays between 0.2 and 1.2 seconds. By the end, average delay converges to approximately 0.35 second.

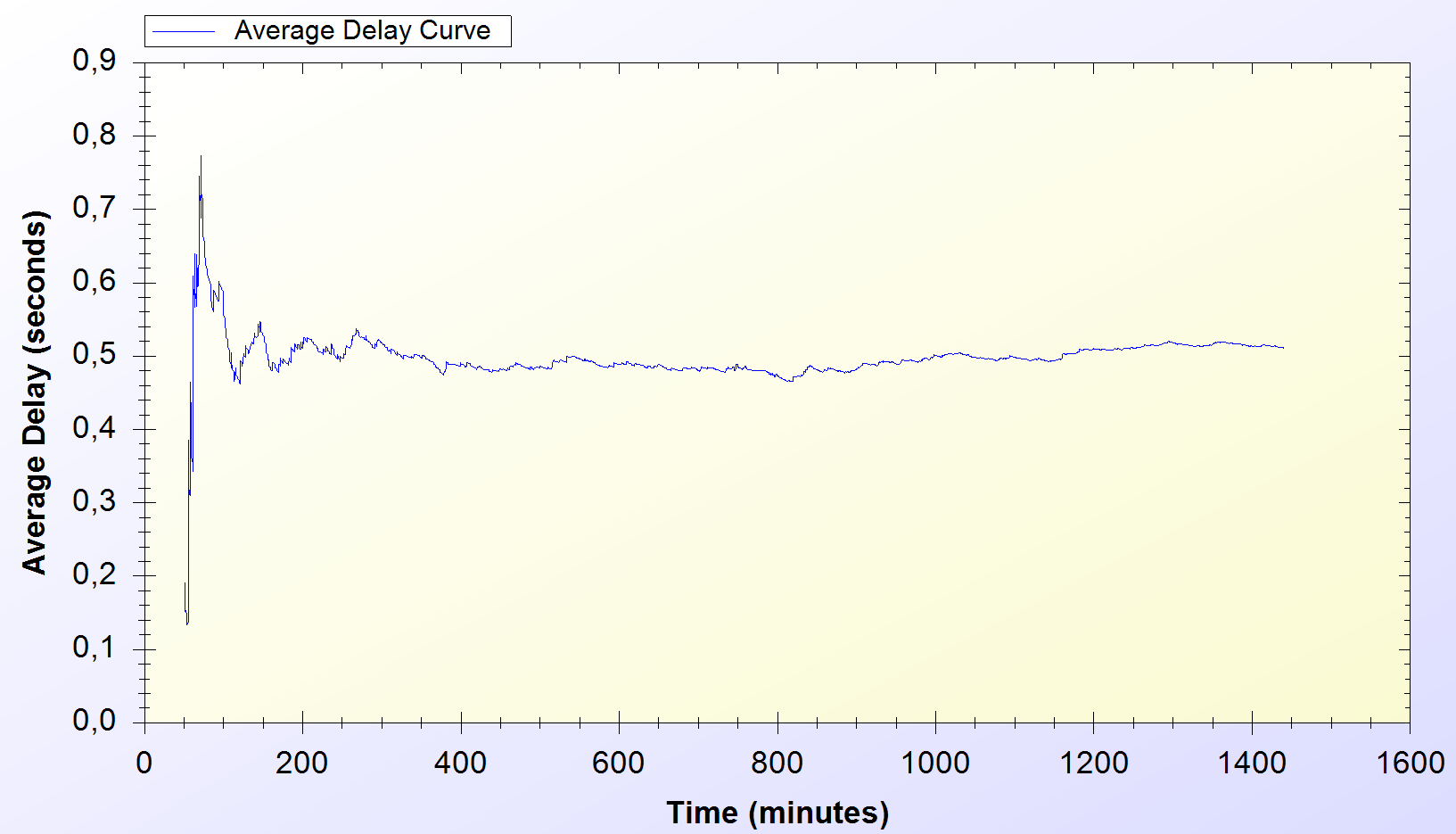


Figure 5.3. Real-Life Simulation Result for Initial Authorization and Reuse-CC Protocols (500 Clients)

Figure 5.3, shows Initial Authorization and Reuse-CC protocols’ performances in a system with 500 clients. In this simulation the protocols show an initial disorder, average delay varies between 0.2 and 0.8 seconds but the steady state performance is achieved after 100 minutes around 0.5 second.

Performance comparison of these protocols is depicted on Figure 5.4. Comparing the results of the simulations with 100, 300 and 500 clients, we could see that network performance slightly decreases as the number of clients increase. However, the decline in system performance stays in reasonable level and shows an increase. The delay values start from 0.3 and gets up to 0.5 second.

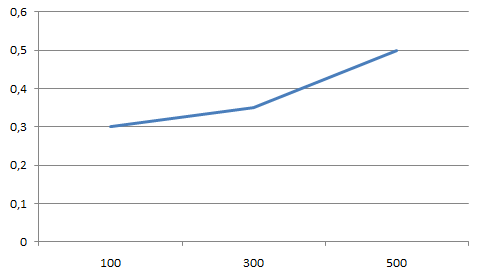


Figure 5.4. Initial Authorization and Reuse-CC Performance wrt. Number of Clients

### Real-Life Scenario Simulation Result for Change Alias

Every active client uses *Change Alias* protocol in the system in every 60 minutes. The protocol is first used at 60th minute and it is used entire time of the simulation.

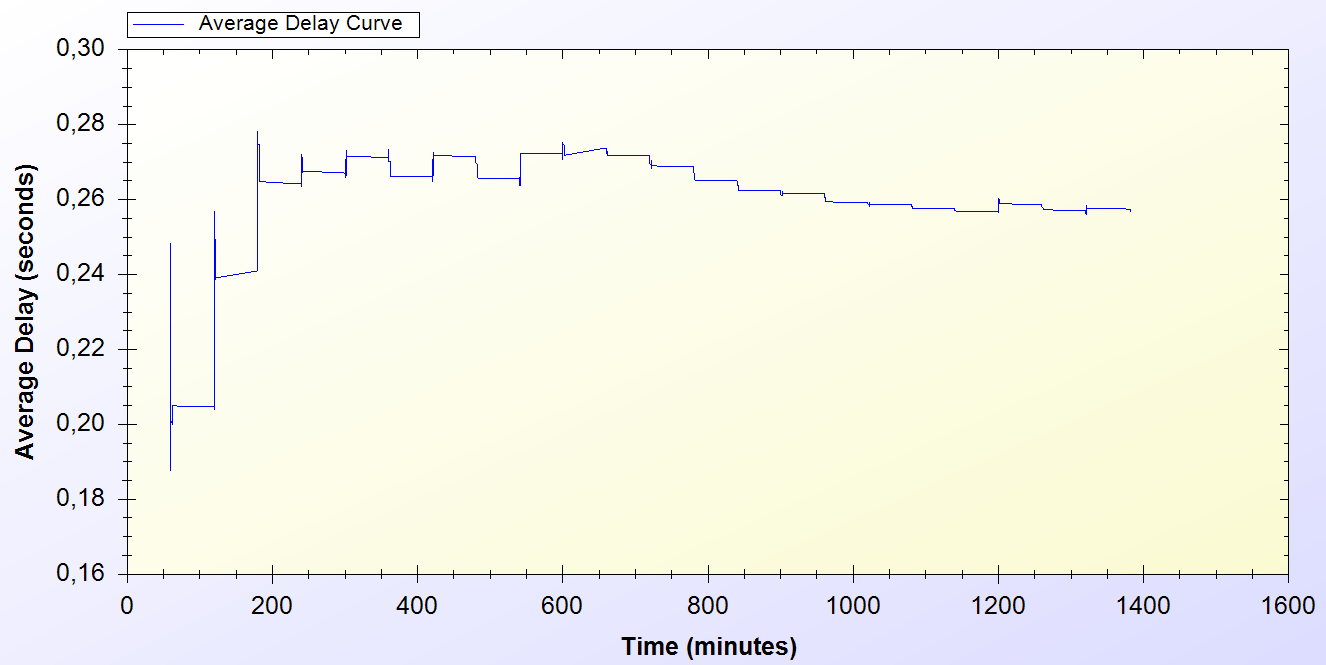


Figure 5.5. Real-Life Simulation Result for Changing Alias Protocol (100 Clients)

Figure 5.5 shows the *Change Alias* protocol performance in real-life scenario simulation with 100 clients. At the beginning of the protocol the delay for the protocol varies between 0.2 and 0.28 seconds. The average delay for the protocol converges to 0.26 second steady state.

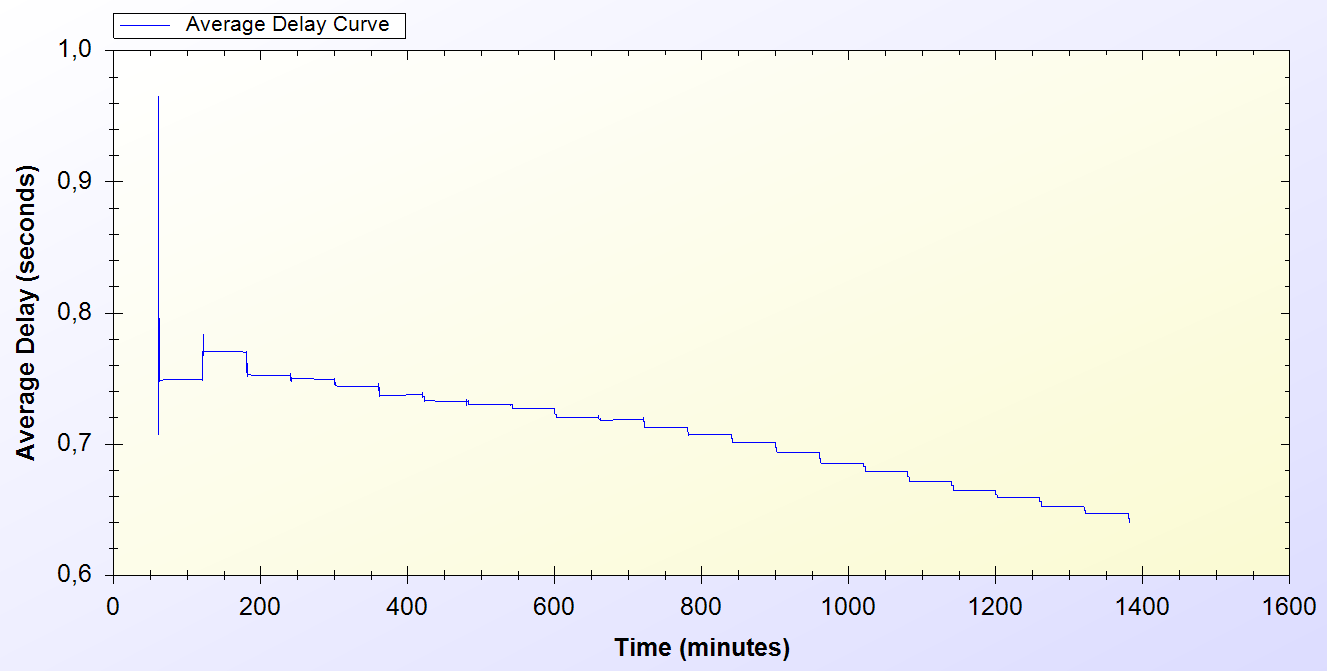


Figure 5.6. Real-Life Simulation Result for Changing Alias Protocol (300 Clients)

Figure 5.6, shows Change Alias protocol performance in real-life scenario simulation with 300 clients. Initial delay values vary between 0.7 and 1 second. The system achieves a steady state performance around 0.65 second after some disorder caused by initial deployment by the system entities. The Change Alias protocol delay values show a significant decline.

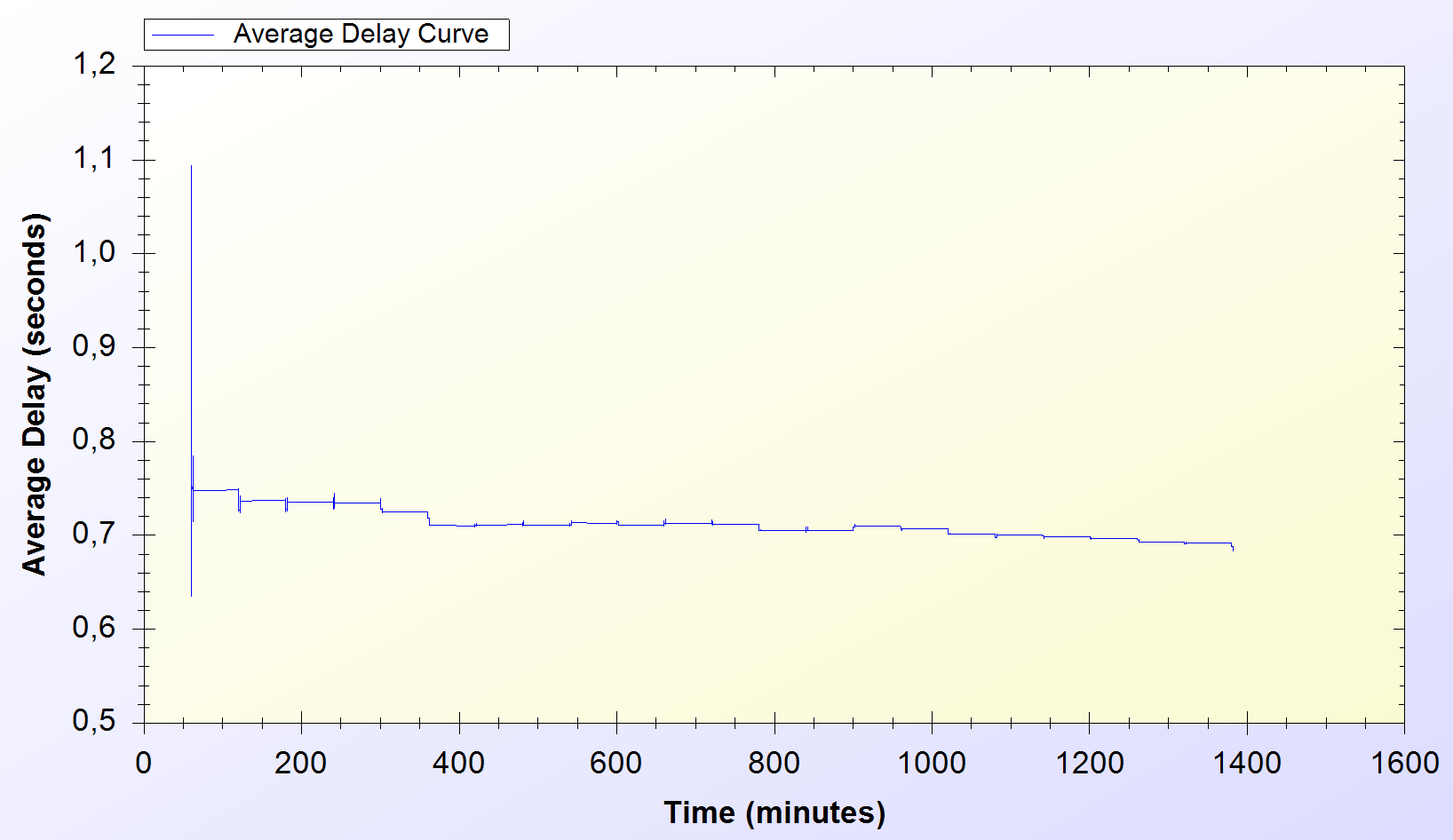


Figure 5.7. Real-Life Simulation Result for Changing Alias Protocol (500 Clients)

Figure 5.7, shows the *Change Alias* protocol performance in real-life scenario simulation with 500 clients. Initial delay values vary between 0.65 and 1.1 seconds. The protocols achieve a steady state delay around 0.7 second.

*Change Alias* protocol has 0.26 second of steady state performance for 100 clients; however, the steady state performance increases sub linearly and reaches up to 0.7 second of delay for 500 clients. The summary of Change Alias protocol performance comparison considering number of clients is depicted on Figure 5.8.

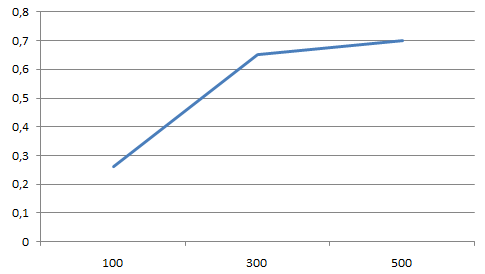


Figure 5.8. Change Alias Performance wrt. Number of Clients

### Real-Life Scenario Simulation Result for Disconnection

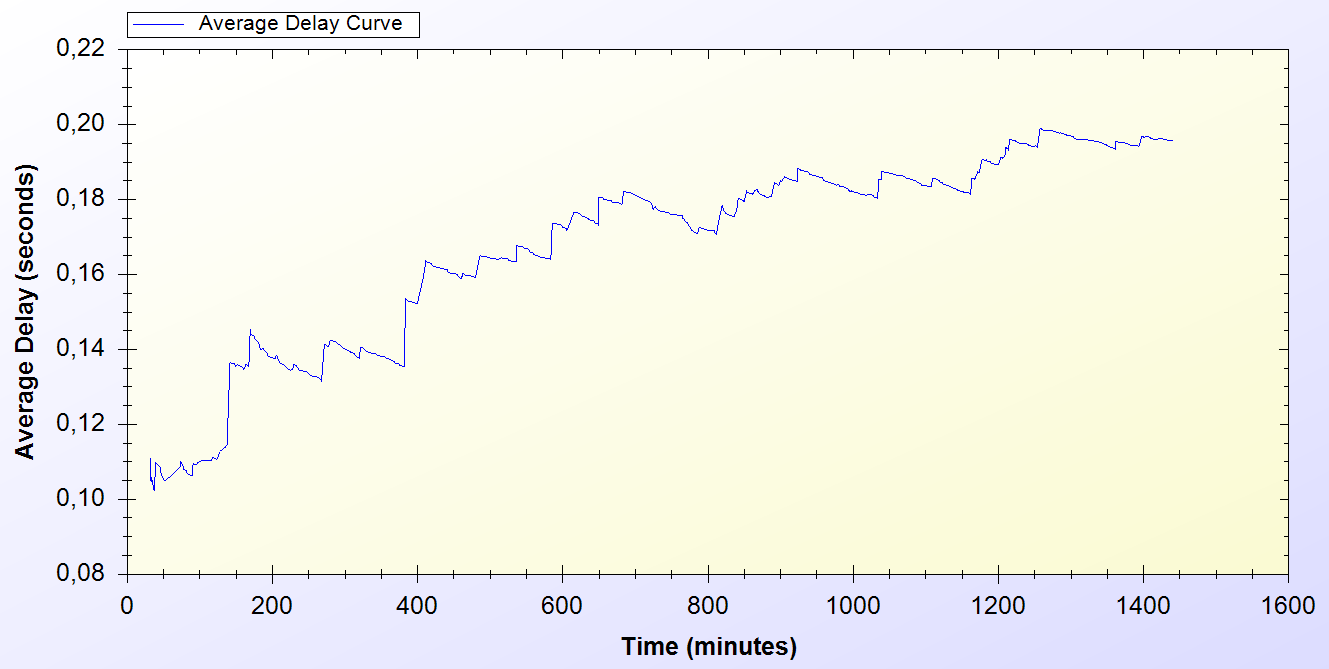


Figure 5.9. Real-Life Simulation Result for Disconnection Protocol (100 Clients)

Figure 5.9 shows Disconnection protocol performance in real-life scenario simulation with 100 clients. Disconnection protocol’s average delay starts from 0.1 second and shows a slight increase by the time passes. By the end the protocol reaches a final performance around 0.2 second.

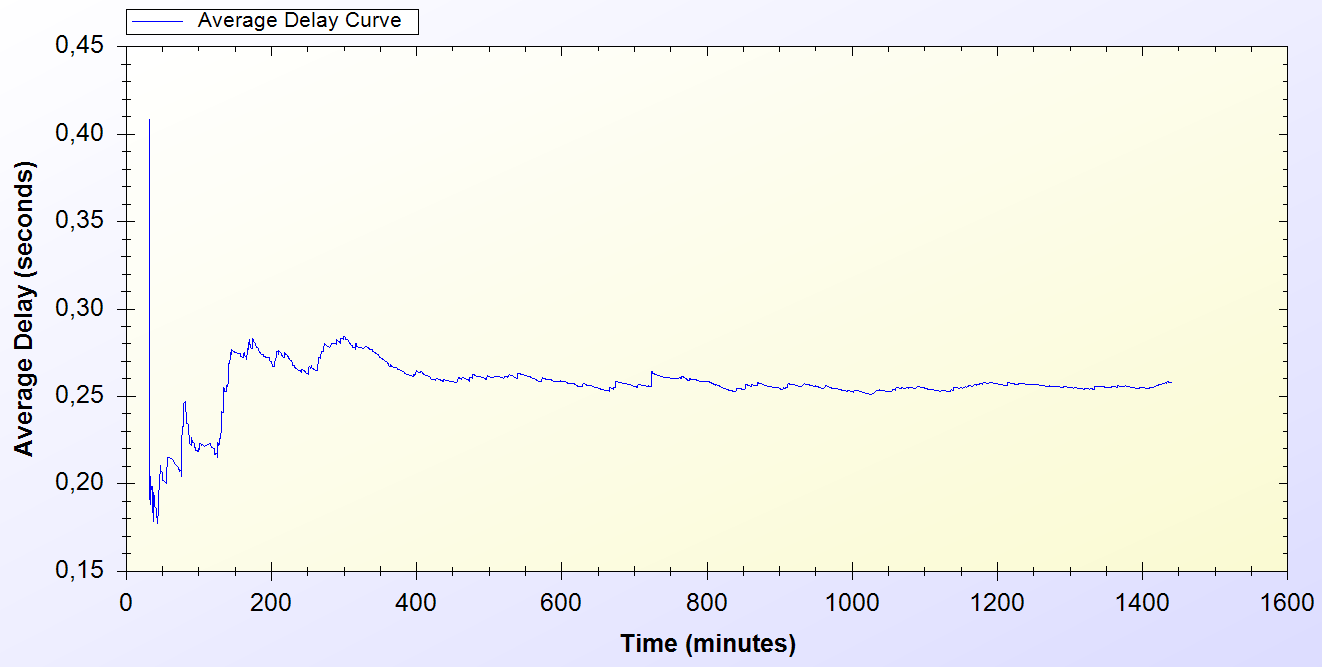


Figure 5.10. Real-Life Simulation Result for Disconnection Protocol (300 Clients)

Figure 5.10 shows Disconnection performance in a system with 300 clients. Initial delay values show variance between 0.2 and 0.4 seconds but then the average delay value converges to a value slightly over 0.25 second.

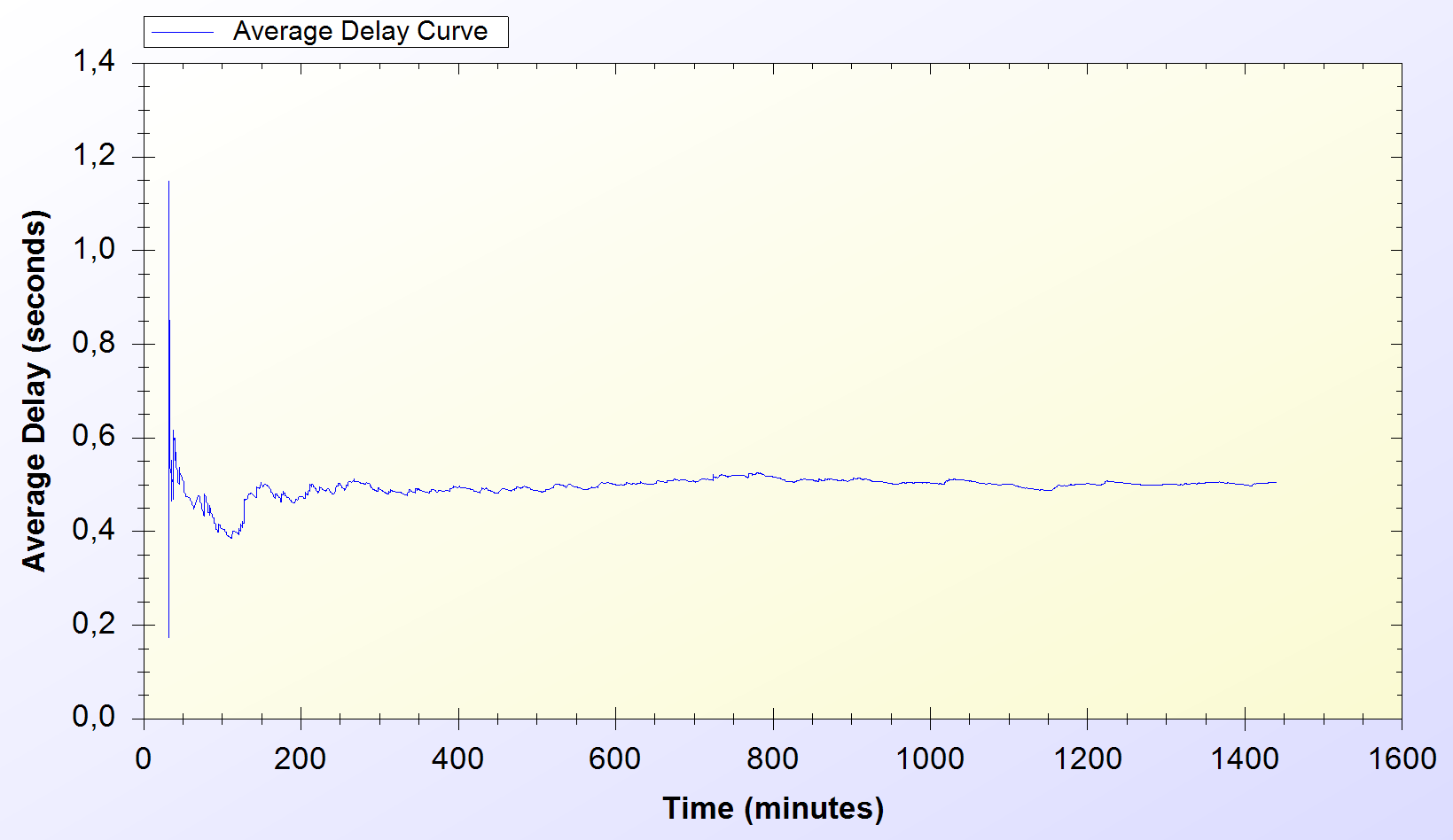


Figure 5.11. Real-Life Simulation Result for Disconnection Protocol (500 Clients)

Figure 5.11 shows Disconnection protocol’s performance in a system with 500 clients. Initially, the protocol delay values differ between 0.2 and 1.2 seconds. Disconnection protocol achieves a steady state performance and causes network and computational delay around 0.5 second.

*Disconnection* protocol shows an increase from 0.2 second to 0.45 second. The protocol shows an increase as the number of clients gets higher. The comparison is shown in Figure 5.12

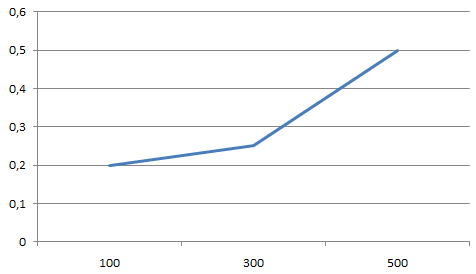


Figure 5.12. Disconnection Performance wrt. Number of Clients

### Real-Life Scenario Simulation Result for Update Packets

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

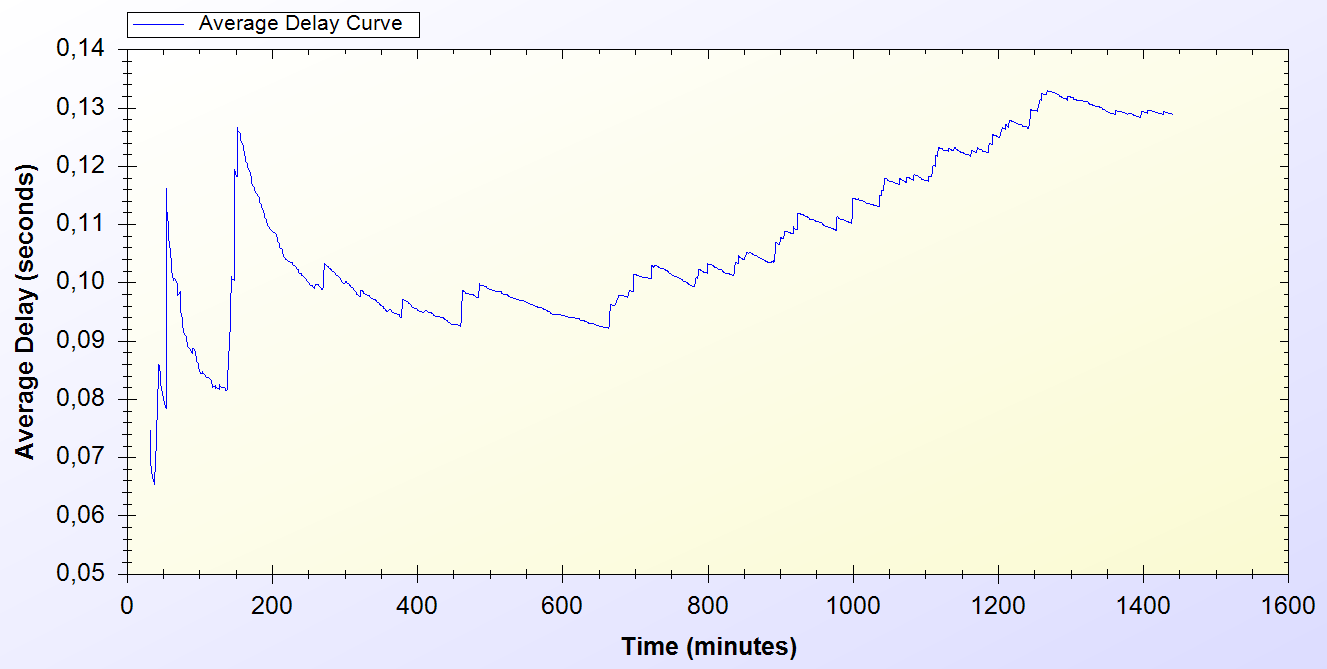


Figure 5.13. Real-Life Simulation Result for Update Packets Protocol (100 Clients)

As it is seen on Figure 5.13, which shows the Update Packets protocol’s performance in a system with 100 clients, at the early stages of the protocol, the average delay value varies between 0.07 and 0.13 second but then after some time the protocol reaches a delay value around 0.13 second.

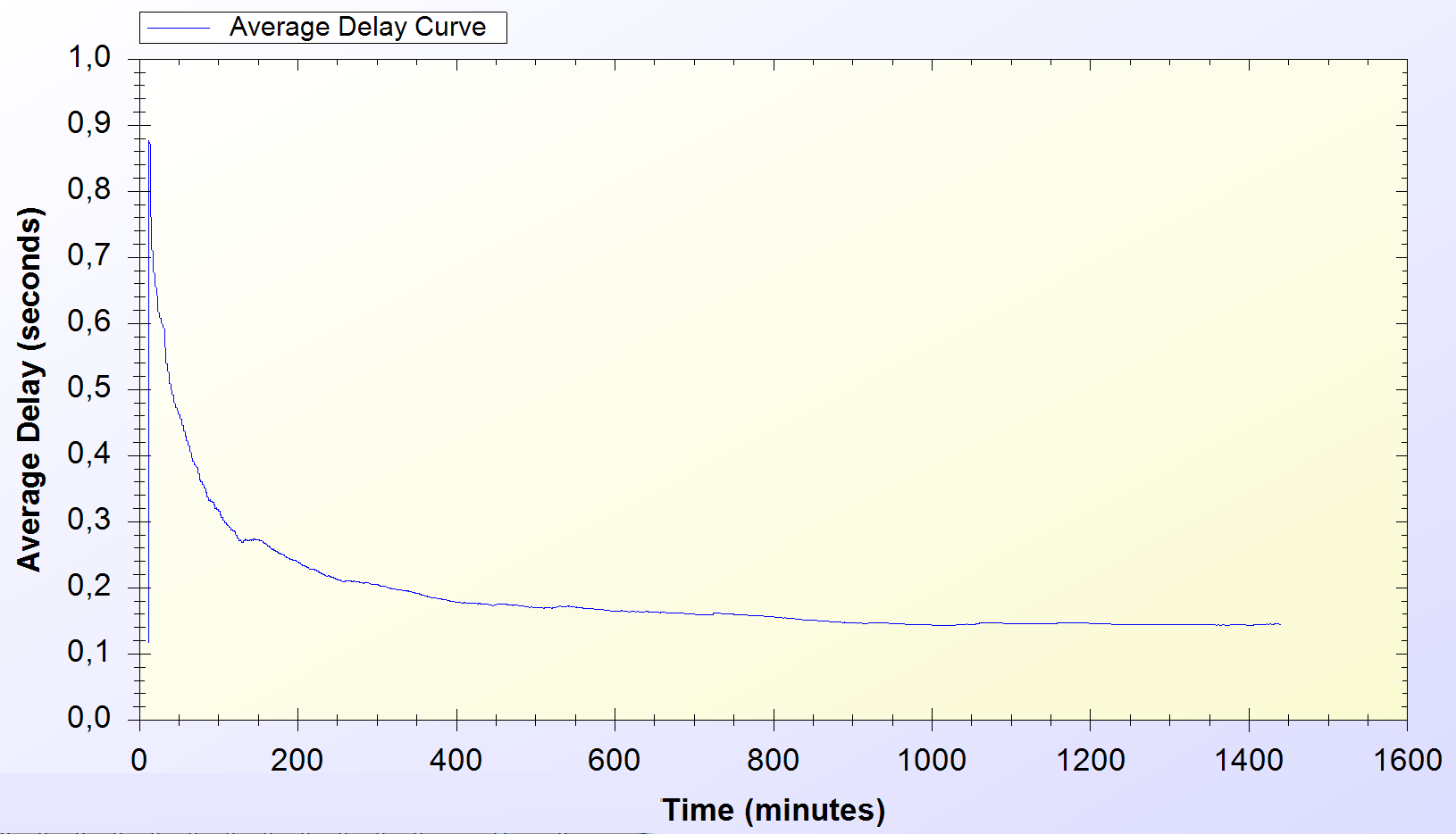


Figure 5.14. Real-Life Simulation Result for Update Packets Protocol (300 Clients)

Figure 5.14 shows Update Packets protocol performance in real-life scenario simulation with 300 clients. The system achieves a steady state performance by the end of the simulation and causes a delay value around 0.2 second.

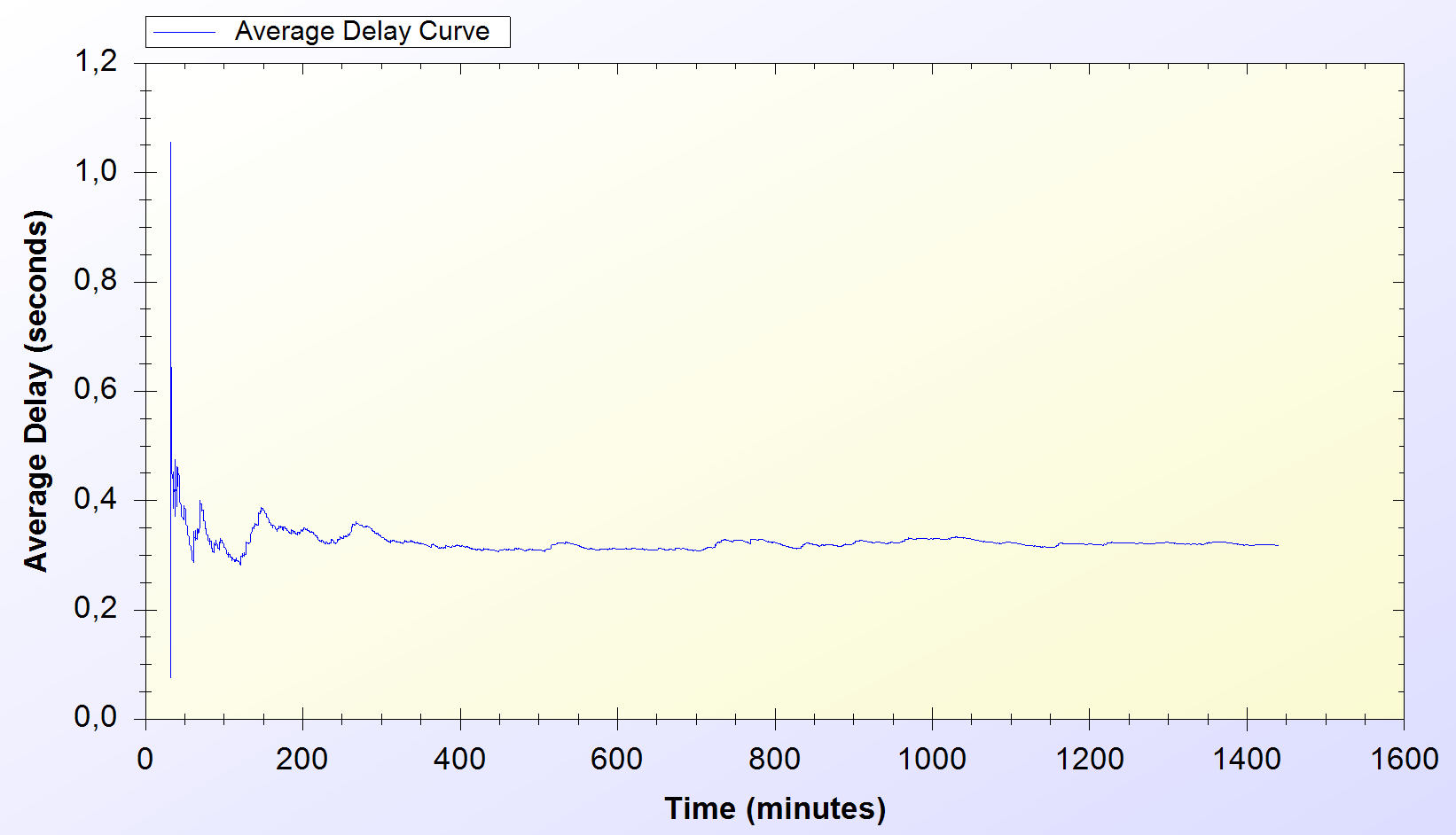


Figure 5.15. Real-Life Simulation Result for Update Packets Protocol (500 Clients)

Figure 5.15 shows *Update Packets* protocol performance under a system with 500 clients. Update Packets protocol achieves a steady state performance around 0.4 second with the previously mentioned settings.

*Update Packets* protocol shows an increase as the number of clients gets higher. The real-life scenario simulation for 100 clients results around 0.13 second of delay; whereas, the simulation for 500 clients gives around 0.35 second of delay. The increase occurs with tolerable values. Figure 5.16 depicts the comparison of Update Packets protocol performance considering the number of clients.

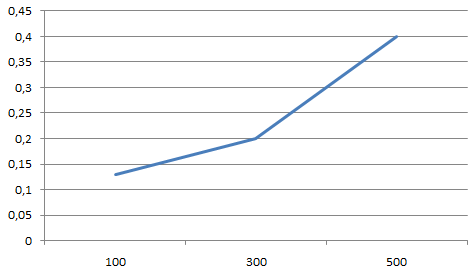


Figure 5.16. Update Packets Performance wrt. Number of Clients

### Real-Life Scenario Simulation Result for Seamless Mobility and Seamless Roaming Protocols

*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.

*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.

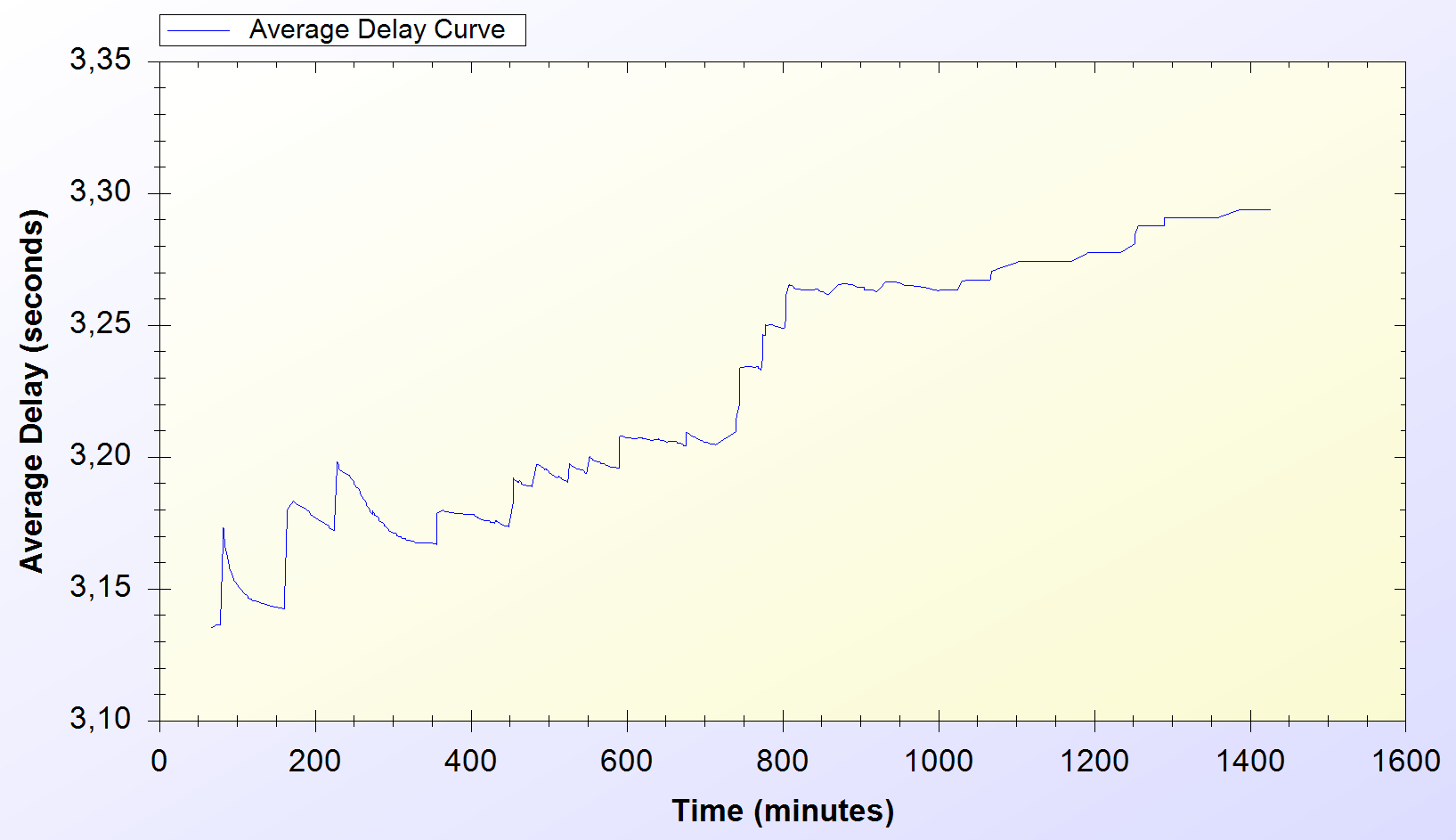


Figure 5.17. Real-Life Simulation Result for Seamless Mobility and Seamless Roaming Protocols (100 Clients)

Figure 5.17 shows Seamless Mobility and Seamless Roaming protocols performance on a system with 100 clients. It could be seen thatthese protocols has an initial average delay of 3.15 seconds and by the time passes the delay that caused by these protocols shows a slight increment. In steady state, a user spends around 3.30 seconds for the overall protocol run. Although this delay seems a bit high for a protocol that aims to be seamless, the protocol execution can overlap the actual usage with old access point. Thus, the client does not stop getting service from the old access point until she finishes all the handover process with the new access point.

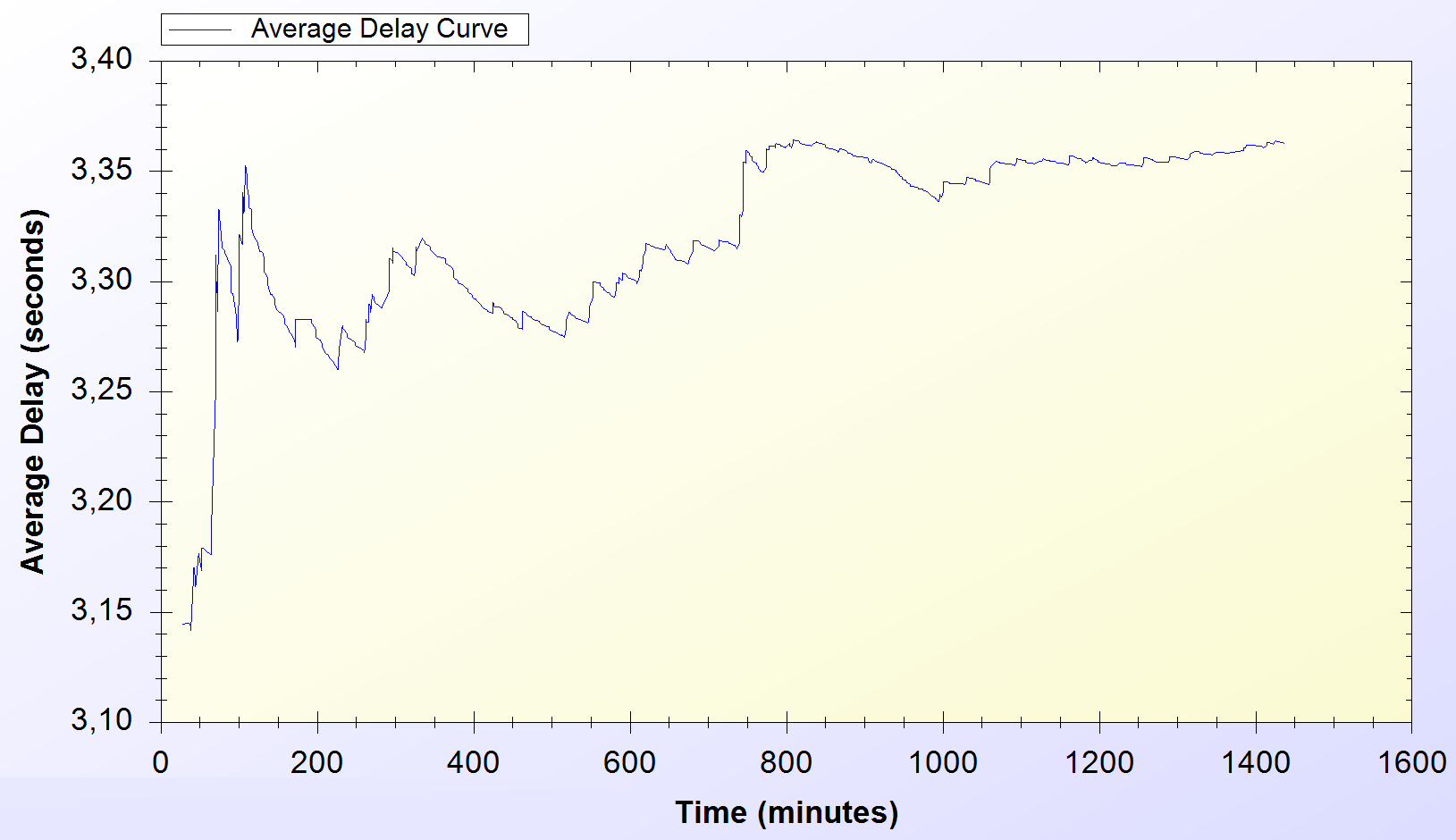


Figure 5.18. Real-Life Simulation Result for Seamless Mobility and Seamless Roaming Protocols (300 Clients)

Figure 5.18 shows the protocols performance on a network with 300 clients. Unlike the ones in previous simulation, the protocols reach a steady state performance around 3.35 seconds of average delay.

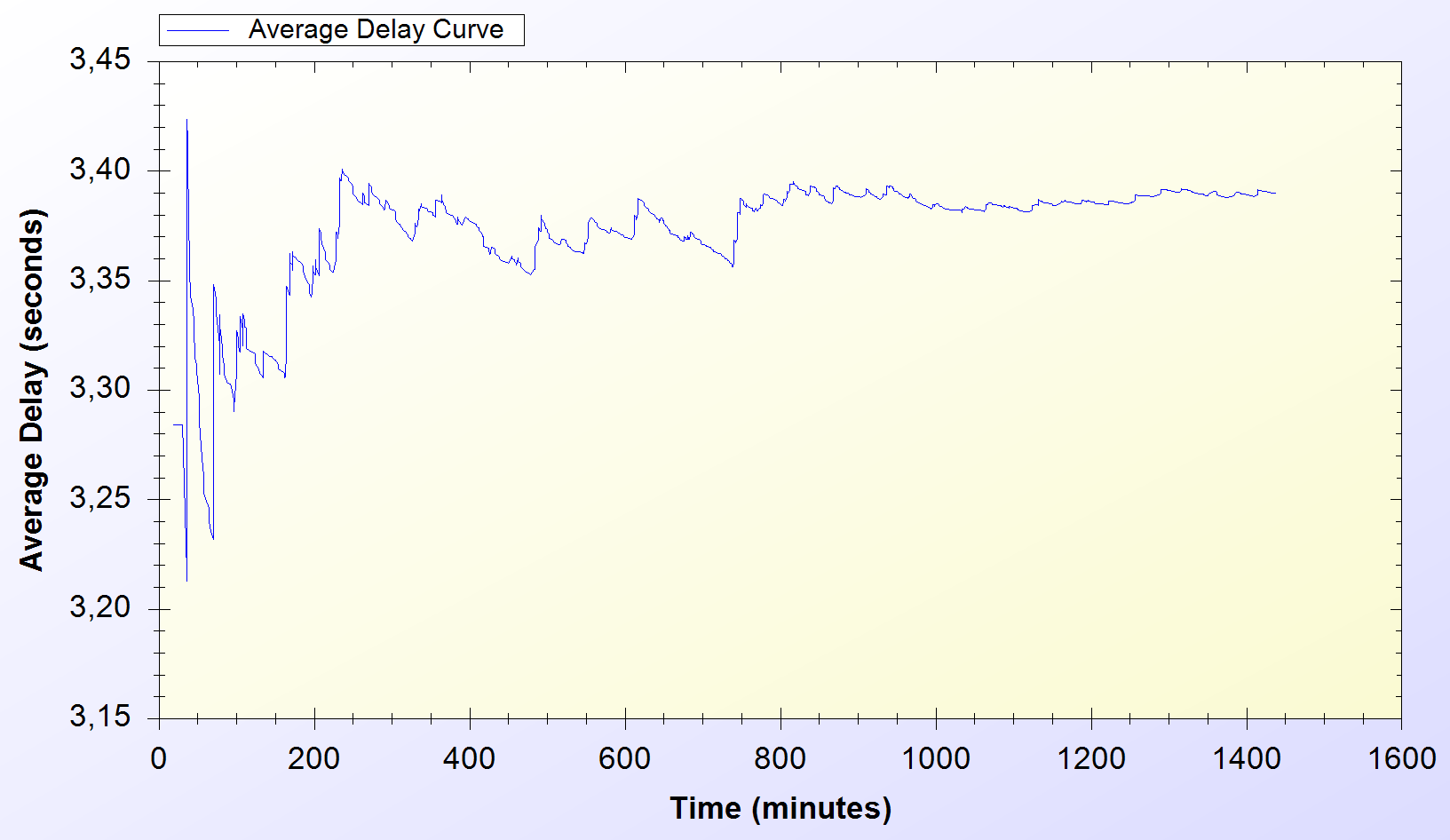


Figure 5.19. Real-Life Simulation Result for Seamless Mobility and Seamless Roaming Protocols (500 Clients)

Figure 5.19 shows Seamless Mobility and Seamless Roaming delays in a system with 500 clients. In this system, the protocols achieve a steady state performance at 3.40 seconds.

Different real-life scenario simulations for 100, 300 and 500 clients show that the delay values for *Seamless Mobility* and *Seamless Roaming* protocols start from 3.3 seconds and reach 3.4 seconds, the increase occurs linearly. These protocols do not show a significant change as the number of clients gets higher. Figure 5.20 depicts the change in the performance of the Seamless Mobility and Roaming protocols performance as the number of clients change.

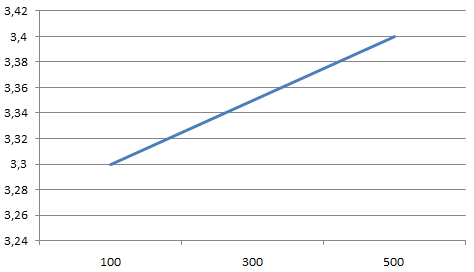


Figure 5.20. Seamless Mobility and Seamless Roaming Performance wrt. Number of Clients

### Real-Life Scenario Simulation Result for Packet Transfer

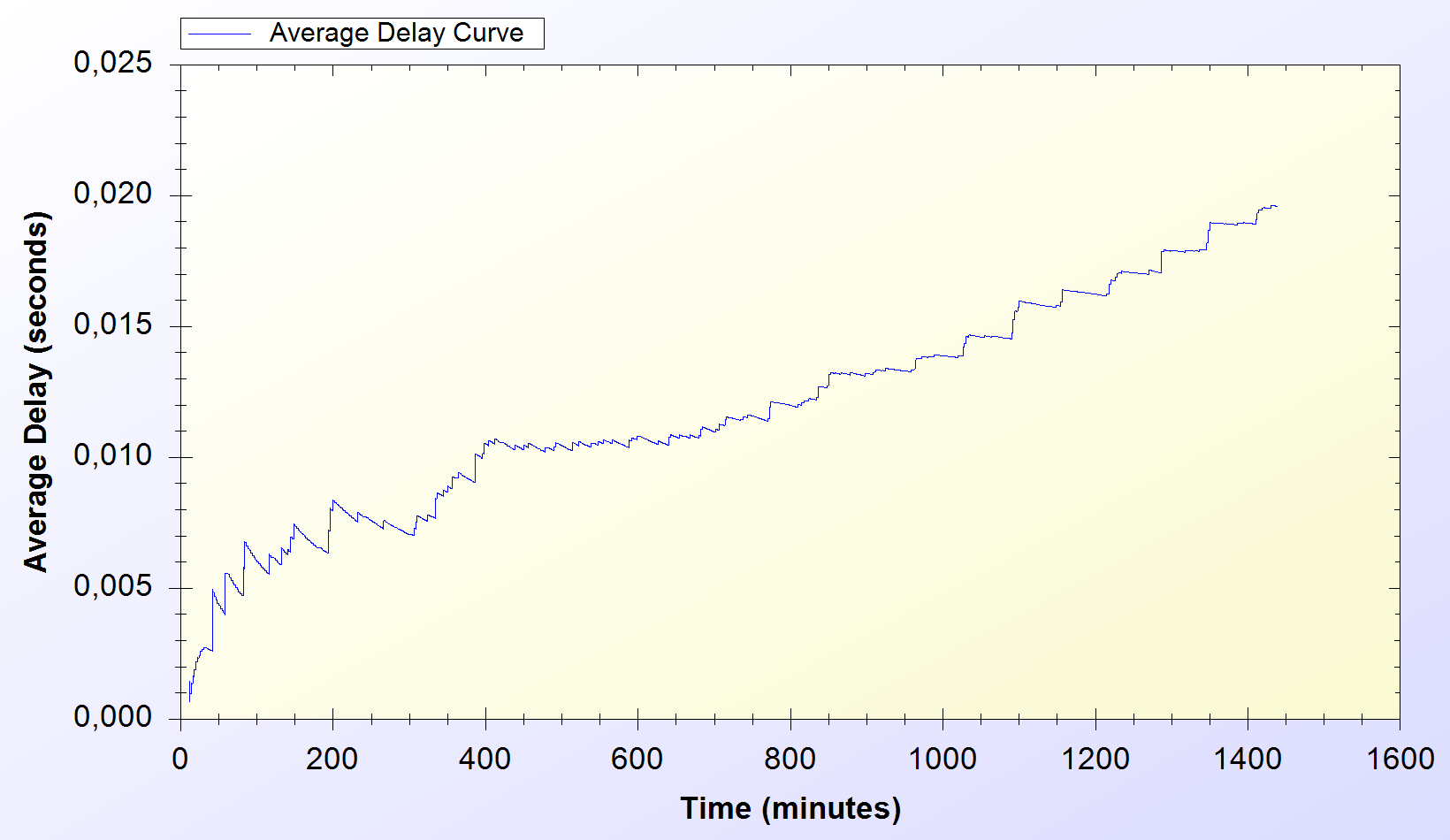


Figure 5.21. Real-Life Simulation Result for Packet Transfer Protocol (100 Clients)

Figure 5.21 shows Packet Transfer protocol performance in a network with 100 clients. The average delay value of the protocol starts from 0.002 second at the beginning of the simulation. The protocol delay linearly increases over time and causes 0.02 second of network delay at the end.

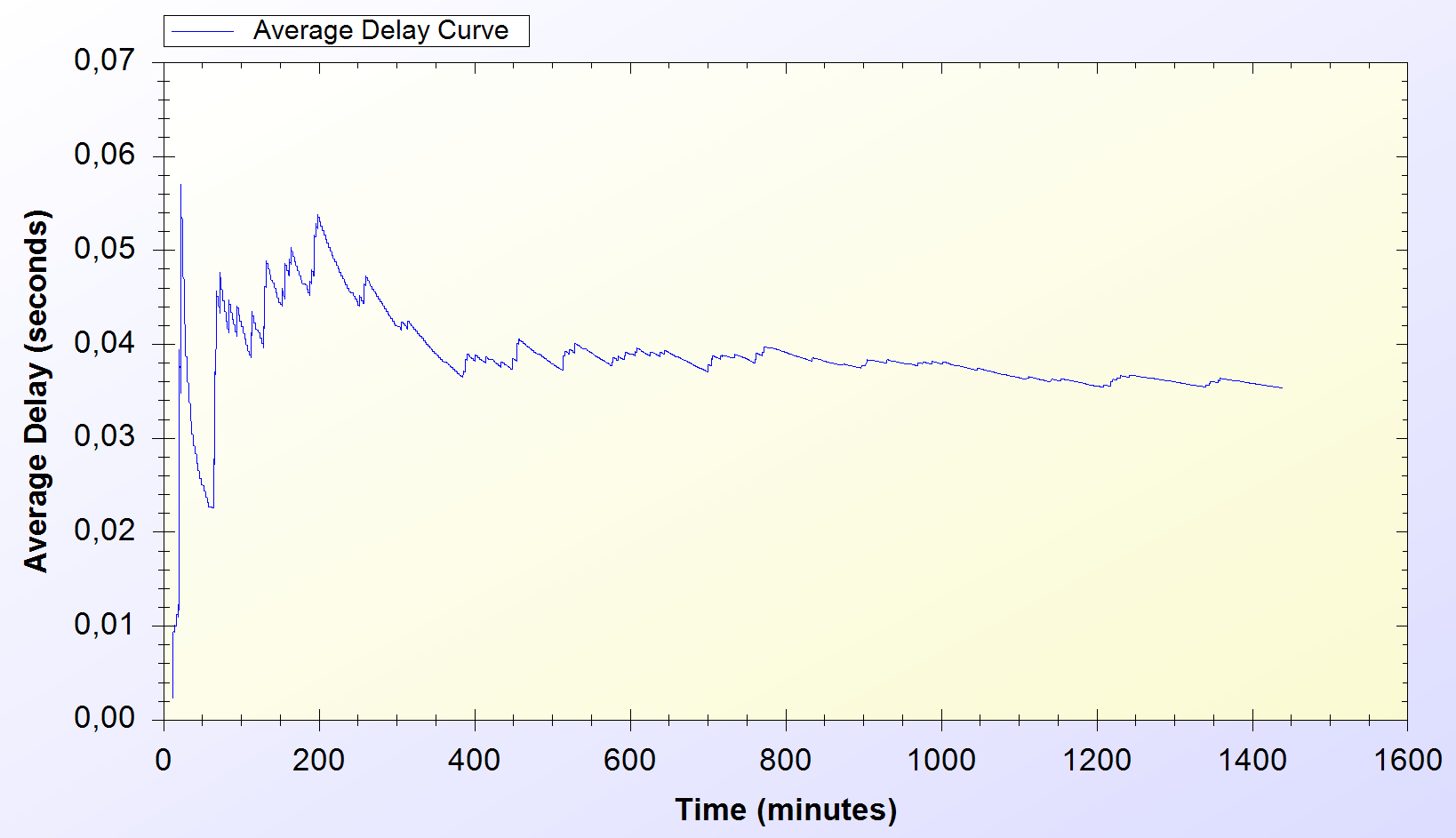


Figure 5.22. Real-Life Simulation Result for Packet Transfer Protocol (300 Clients)

Figure 5.22 shows Packet Transfer protocol performance in a network with 300 clients. Initial performance of the protocol is instable but it achieves steady state around 0.04 second of average delay.

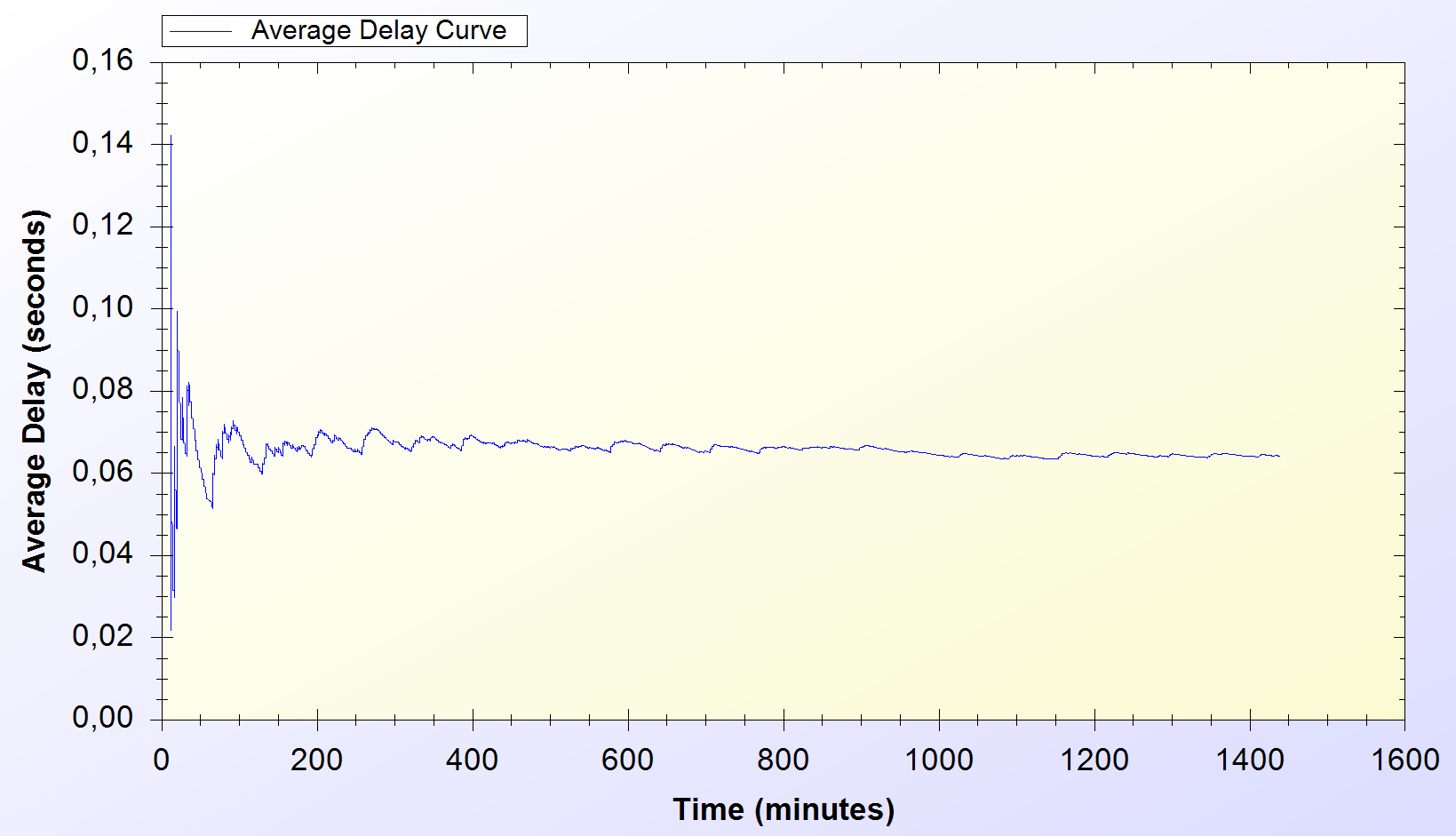


Figure 5.23. Real-Life Simulation Result for Packet Transfer Protocol (500 Clients)

Figure 5.23 depicts the performance of Packet Transfer protocol in a system with 500 clients. The protocol achieves a steady state average delay around 0.06 second.

Packet Transfer protocol delay linearly increases as the number of clients increase. The protocol delay is 0.02 for 100 clients; whereas, it is around 0.06 second for 500 clients. Figure 5.24 shows the change in Packet Transfer performance as the number of clients increase.

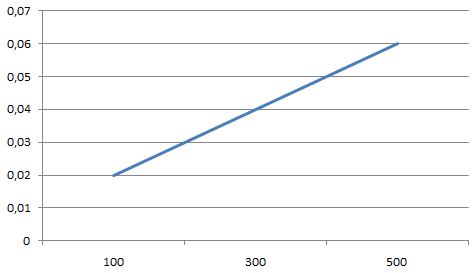


Figure 5.24. Packet Transfer Performance wrt. Number of Clients

### 4.4.2. Overall Burden of the System

Real-life scenario simulation for 500 clients provided the results in Table 5.1. Charts on Figure 5.25 and Figure 5.26 are drawn exploiting the results in Table 5.1 In average; over 1000 minutes of Internet service needs a delay of 12 to 15 minutes of waiting; which yields approximately 1.5% overhead.

Table 5.1: Simulation Results for Client Types

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Total Internet Usage Time | Total SSPayWMN Delay | Average Internet Usage Time for a Client | Average SSPayWMN Delay for a Client |
| Students | 160858 Minutes | 2660 Minutes | 963 Minutes | 15 Minutes |
| Employees | 168296  Minutes | 2062 Minutes | 1013  Minutes | 12 Minutes |
| Domestics | 176792 Minutes | 2558 Minutes | 1058 Minutes | 15 Minutes |

Difference between client types affects the system usage of the clients. The probabilistic values, which are mentioned in Section 4.2, determine the system usage frequency of the clients. As it is seen on Table 5.1, domestics are the most active clients in the system. Employees and students follow domestics in that sense. On Table 5.1, the simulation results are grouped into two subgroups, which are: Total Internet Usage and Average Internet Usage. Total Internet Usage means the overall sum of network usage of these 500 clients in a day. When we analyze the delay values we see that a client experiences a delay, which is approximately 1.5% of the total amount of received service.

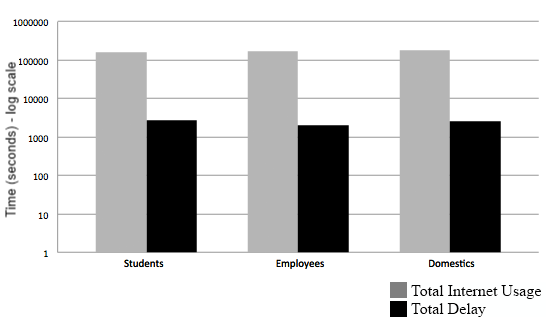


Figure 5.25. Total Amount of Service Usage Times for Client Types vs. Total Delays

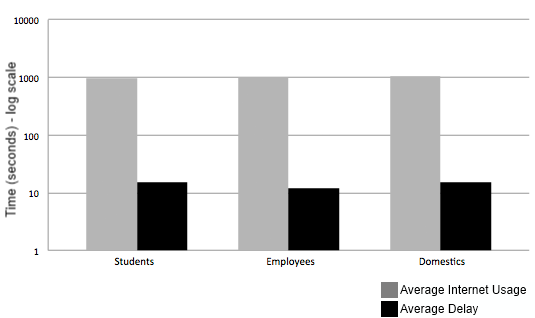


Figure 5.26. Average Service Usage Times for Client Types vs. Average Delays

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

Figure 5.25 and Figure 5.26 shows the overall results for real-life scenario simulation. Figure 5.25 shows comparison of minutes clients used as idle or active. Figure 5.26 shows the average value for the clients of the same group.

# Conclusion

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

The performance of SSPayWMN has been evaluated with simulations using ns-3. Two groups of simulations are performed: unit tests [4] and real-life scenario simulations. Unit tests ensured the stable performance of the protocols in stand-alone run; whereas, real-life scenario simulations ensured the stable system performance in real-life situations. We have conducted real-life scenario simulations for different number of clients: 100, 300 and 500. We have compared the protocol performances considering the change in clients count. The difference between the average delay values of different real-life simulations showed that with increasing number of clients SSPayWMN protocols show higher network delays. However, the increase is linear; therefore, SSPayWMN ensures stable performance in different sized networks

There are different user types, as there are different types of clients in real life. There is also randomness at the system, so we have different outcomes for the same simulations as there is change in network traffic everyday despite the users are same. 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.

Our results show that the SSPayWMN system works in very reasonable amount of time. It is secure and fast. The simulation results are satisfactory indicating that the SSPayWMN works effectively and it is a respectable system.

**References**

*[1] Levi, Albert Leloğlu, Can Serhat Requirements Specification Document 16.08.2011*

*[2] Levi, Albert Leloğlu, Can Serhat Simulation Tools Comparison Document 25.10.2011*

*[3] Levi, Albert Leloğlu, Can Serhat Protocol Design Document 04.04.2012*

*[4] Levi, Albert Leloğlu, Can Serhat Unit Simulations Report 30.09.2012*