ENSC 427: COMMUNICATION NETWORKS

SPRING 2019

FINAL REPORT

Analysis of Gaming Using Peer to Peer Paradigm

http://www.sfu.ca/~mbin/427project.html

Name	**Student ID**	**Email**
XueQing (Molly) Bin	301222893	mbin@sfu.ca
PengCheng (Liam) Li	301208101	pla68@sfu.ca
JiaHui (Mandy) Xiao	301206181	mandyx@sfu.ca

TEAM #6

Table of Content

List of Figures

1. ABSTRACT

The internet gaming industry has quickly grown to be one of the predominant pastimes for people of all ages and in recent years have seen the development of numerous professional Esports leagues with significant financial backing. A quality implementation of internet gaming is a resource intensive task as the timing requirements are held to a high standard. This project is a comprehensive study on internet gaming using P2P paradigm where a peer sends and receives packet information from other peers. It also dives into the benefits of P2P paradigm such as reduced lag and issues surrounding security. All network simulations is performed in Riverbed.

2. INTRODUCTION

As online games become increasingly resource intensive, center coordination by servers are not sufficient to provide satisfactory timing requirements. Latency problem has affected gaming experience as lags could severely disrupt ongoing games. We need to develop games that takes into consideration factors such as consistency, responsiveness, reliability, security and persistency. Since client-server gaming network is expensive to operate and the breakdown of one component risks breaking down the whole system, it tends to have a large latency as a server can be easily bottlenecked when several client's requests for services. We have examined an alternative that eliminates these issues: peer-to-peer (P2P) gaming network.

P2P network is a network such that two or more *peers* are computer systems connected to each other via the internet. In a P2P system. Files therefore can be shared directly between each client without the need of a central server.

Since more and more P2P games has been developed in recent years, in this project, we will build and simulate through Riverbed to examine the multiplayer online game performance of P2P architecture by facilitates changes in number of players and size of packets. We will evaluate simulation outputs such as traffic sent, traffic dropped, end to end delay and response time.

2.1 Related Works

Research around areas related to P2P Massively Multiplayer Online (MMO) games have became more prevalent in recent years. Some of these related works include *Designing a Super-Peer Network* [7]. It is mentioned in the paper that varying types of P2P systems have different levels of centralization. There are three main types of P2P systems. In a pure system, equal amounts of traffic and responsibilities are distributed amongst all peers in the network. In a hybrid system, only the download actions are P2P, otherwise it utilizes the Client/Server model. In the Super-Peer system, it uses the super-peers as a centralized server for other peers and their clients. Since this system is a combination of the pure and hybrid, it takes on the advantages in both such as autonomy, load balancing and robustness to attacks [7].

Using P2P systems for MMO is still a relatively new and evolving topic, so some researchers not only explore the benefits in comparison to more traditional paradigms, but also spend time to point out the challenges in P2P gaming. A research paper related to this topic titled *Challenges in Peer-to-Peer Gaming* [6] looks at the challenges that comes with this promising new paradigm. Some of the possible issues discussed are concerns with the distribution of storage amongst millions of players at a time. This leads to problems with the consistency of the game states, to which it is crucial to these fast-paced MMO games. Furthermore, researchers are yet to determine the optimal algorithm for the selection of peers to communicate with. These are some of the interesting works that can be explored surrounding the P2P MMO topic.

3. MAIN SECTION

3.1 Problem Definition

Yahyavi and Kemme [4] analyze various network architectures in online gaming, and comparing the strength and weakness of those architectures. The paper contains explanation of general concepts involved in designing a multiplayer game using client-server and P2P architectures. In this project, we only concentrate on P2P. The Peer-to-Peer architecture as shown in the Figure 3.1.1, increased scalability from the traditional client-server architecture since every node acts both as a server and a client. With P2P, the load is distributed among all nodes and managed by the peers, where the data is managed by each peers. Hence the need for expensive server is reduced and the risk of server breakdown is eliminated.

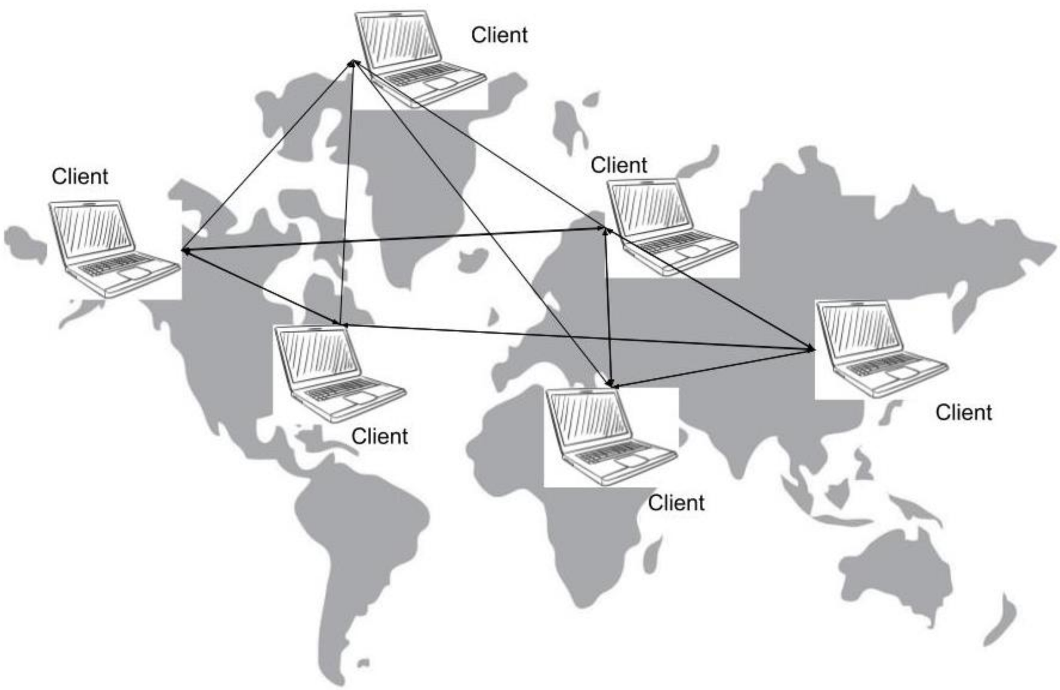


Figure 3.1.1: P2P Topology

3.2 Overall Design

To efficiently analyze the effects of varying packet size and number of peers have on the efficiency of P2P gaming, we have designed three different scenarios. The first scenario is a simple base case consisting of three identical peers each connected to their own routers in a network. In the second scenario, we increase the packet size in order to observe the effects on the overall system. In the third scenario, we vary the number of peers in the network. Each scenario is explained in greater detail in the following subsections.

Before we dive into each scenario, we must describe precisely the *peers* in our network. Figure 3.2.1 is a finite state machine for one of the peers in our overall design. The peers are put into the overlay network and configured so that specific resource requests can be directed to connected peers. In each case we are sending resource requests relating to the current online gaming session. When a neighboring peer with the desired resource is in the off state and thus no longer able to provide resources, the overlay network will redirect the request to the next appropriate peer.

In the Requesting state, a peer will send requests to adjacent peers. Following that is the Waiting state, where the sender waits for response from adjacent peers until the desired response is received. Finally, that peer will move into in the Processing state, where it processes the response it has received and returns to the Waiting state after the processing has completed.

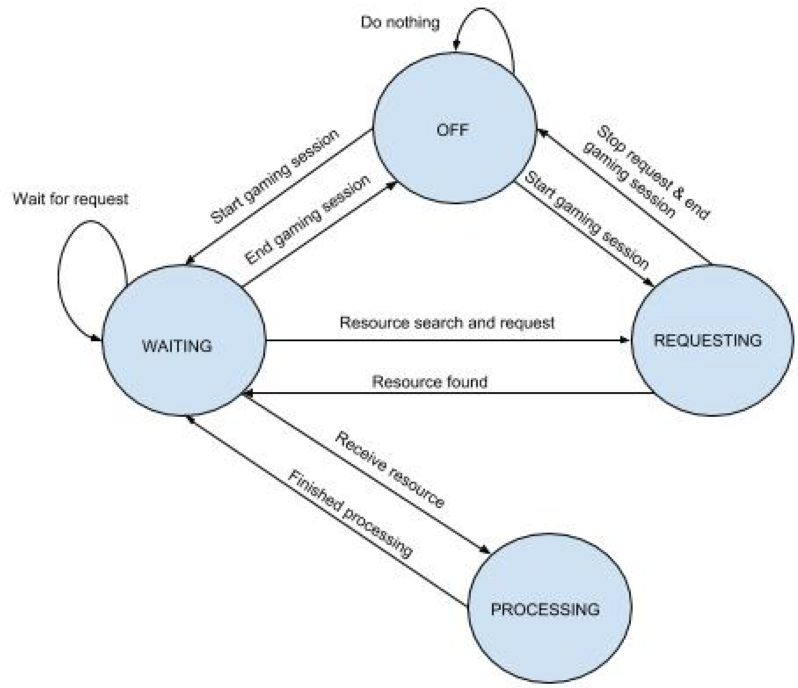


Figure 3.2.1: State Machine of a Single Peer

3.3 Simulation Setup and Implementation

Our project aims to analyze the peer-to-peer paradigms and we decide to use Riverbed Modeler to perform our simulations. The peer-to-peer paradigms consists of three different scenarios. First scenario which is the baseline operates in normal conditions. Second scenario will be simulated with increased packets size. And the third scenario is simulated with three additional clients. For the first and second scenarios, we use three peers. We analyze those three peers across all simulations, Peer1, Peer2, and Peer3. For the third scenario, we use six peers for analyzing their performances. For each scenario, we generates 100BaseT 100 Mbps links between the peers and the routers. PPP_DS3 44.6Mbps links between the router and the internet node. All scenarios are simulated for 40 minutes based on a typical length of time spent per game session.

Figure 3.3.1a shows the network topology of the peer to peer simulation layout that was used for first and second scenario.

3.3.1 First Scenario

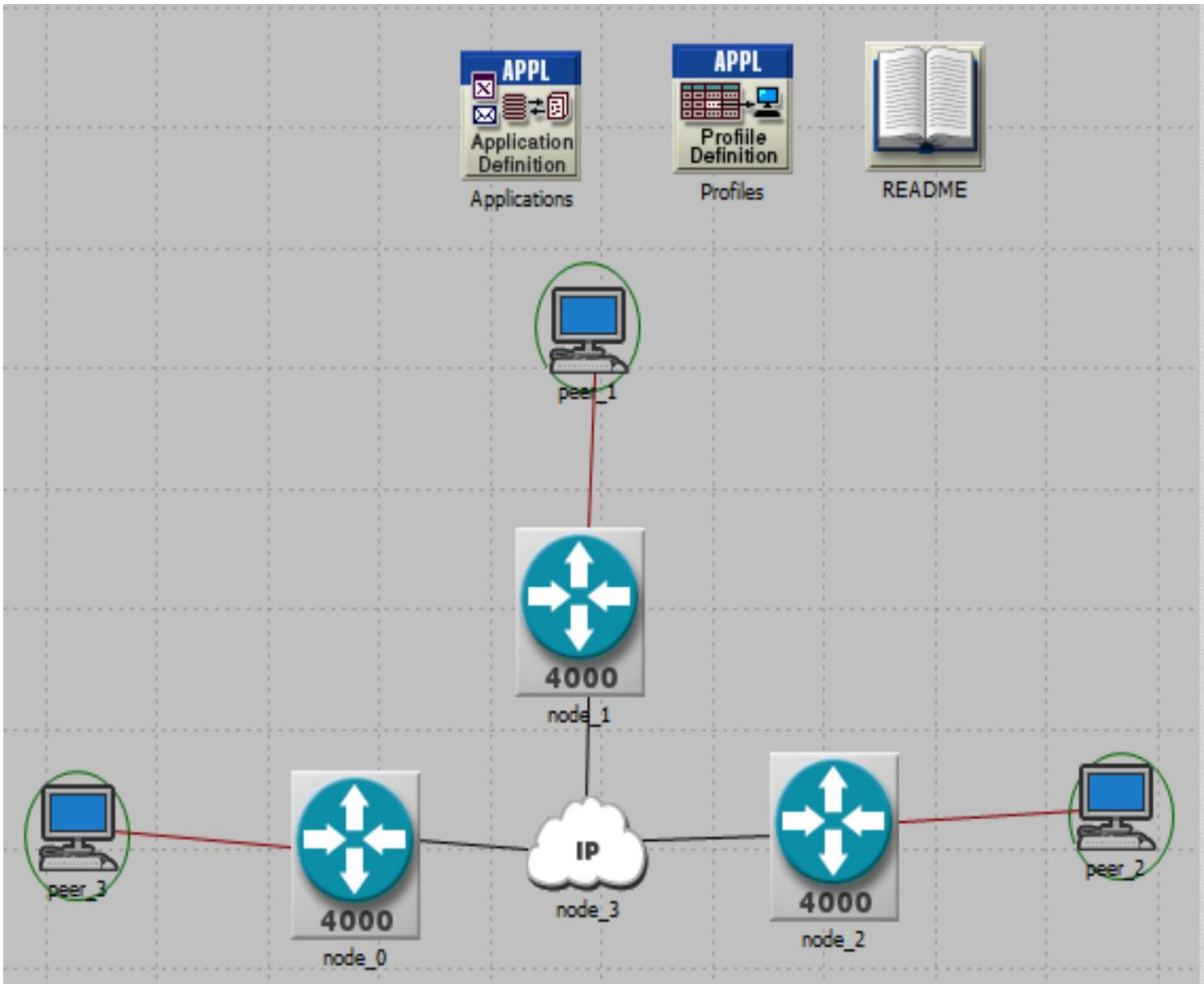


Figure 3.3.1a: network topology for simulation

A modified version of an inbuilt application called Peer-to-Peer-File-Sharing was used to ran our simulation for all nodes. Figure 3.3.1b shows the detail setting of application and profile attributes that was used for the first scenario. The modified application, making it suitable for testing peer-to-peer based online gaming. We modified the packet size to be constant in order to analyze our average rate of the IP traffic dropped. In first scenario, the packet sizes are limited to 80 bytes, and the inter-request time is set to a mean of 0.25 minutes in a Poisson distribution.

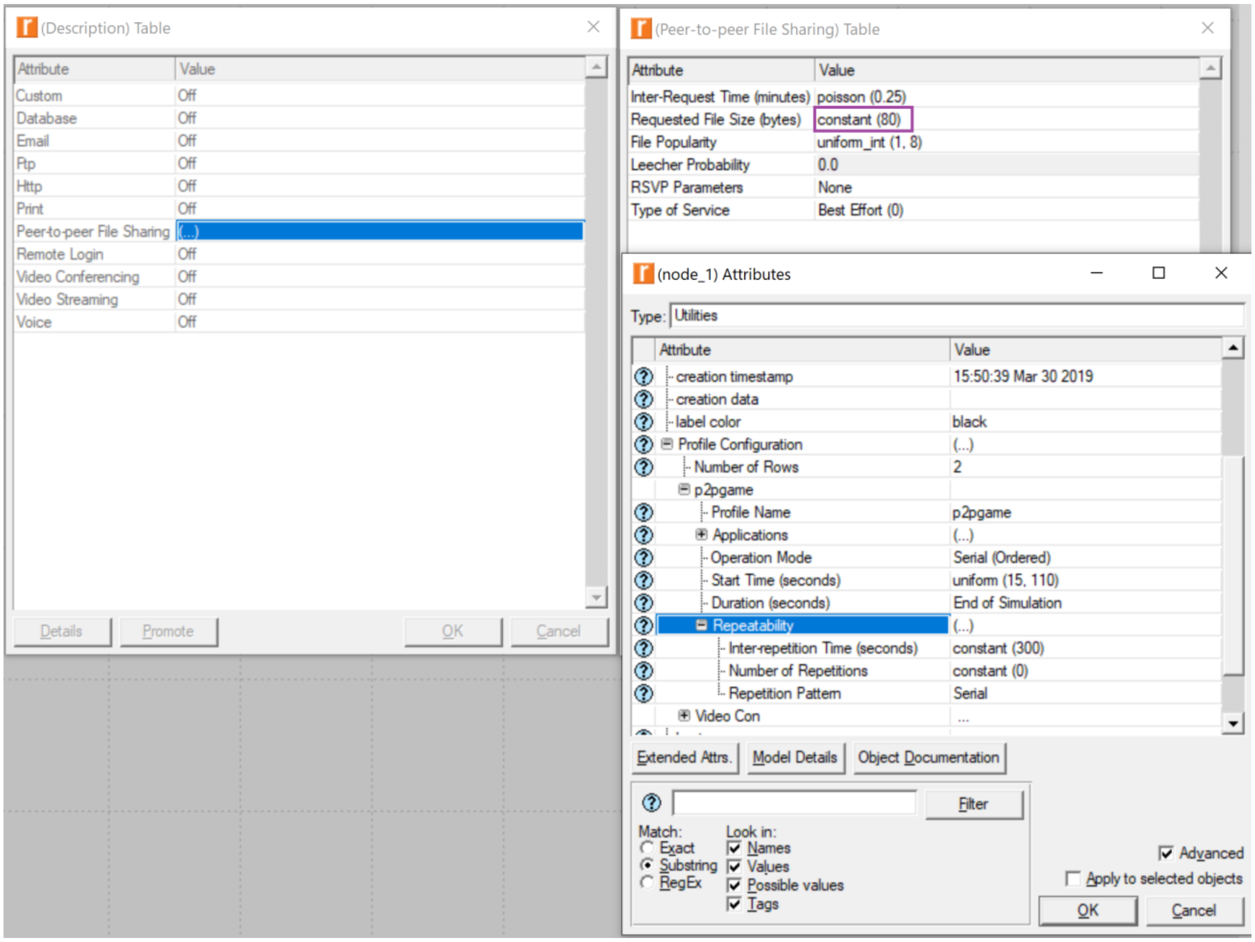


Figure 3.3.1b: Peer-to-peer application and profile setting

3.3.2 Second Scenario

In the second scenario, peer-to-peer topology is kept the same as shown in Figure 3.3.1a. The application setting and profile attributes that were used to run the second scenario are shown in Figure 3.3.2 below. The only parameter which differs is the request file size to which we have increased from 80 bytes up to 8000 bytes. By changing a parameter, the packet size, it is clearly showed how it affects the peer-to-peer performance. This is demonstrated in the analysis of the difference for simulation results including the Download Response Time measure, average traffic sent/dropped and end-to-end delay violation for each peer nodes.

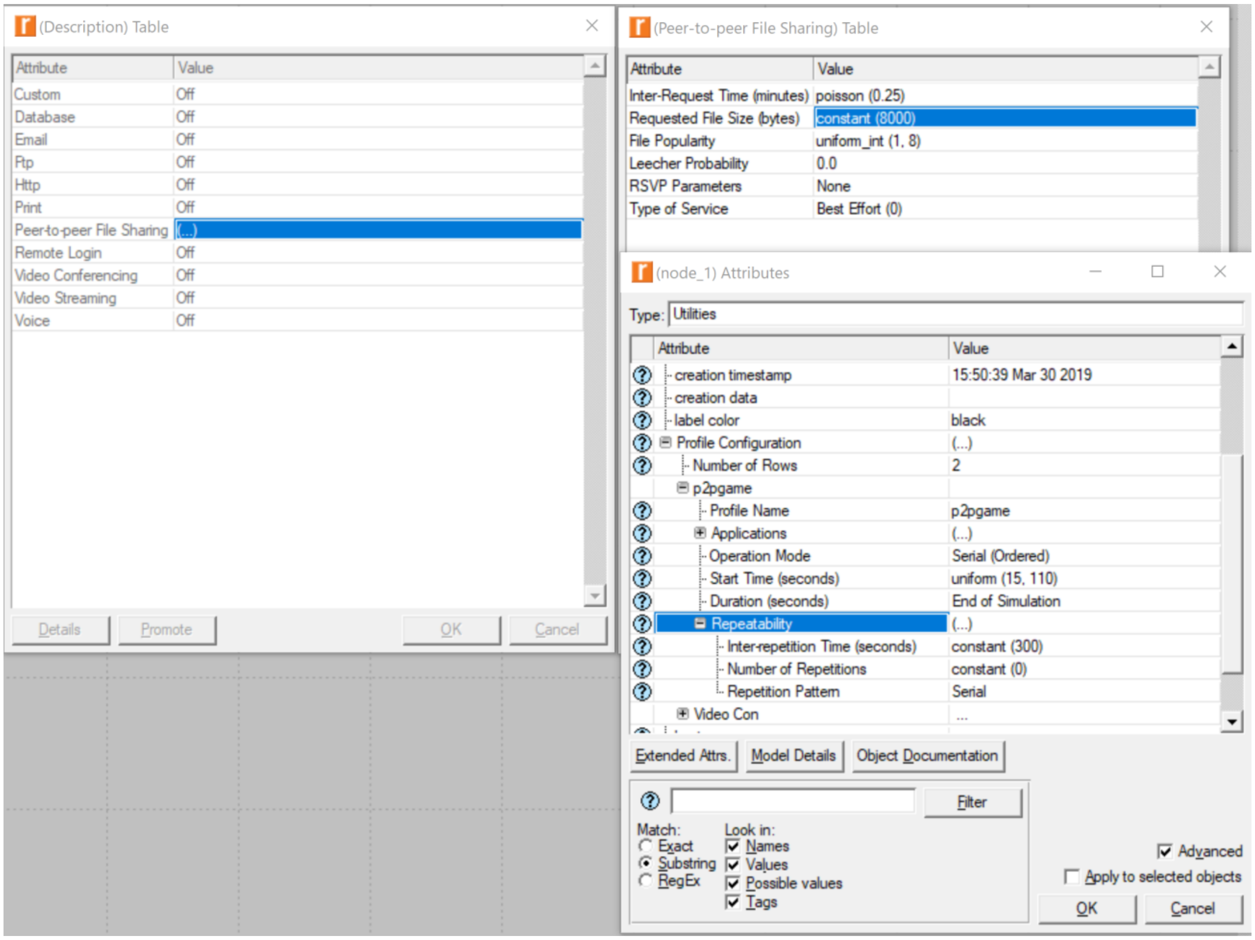


Figure 3.3.2: Peer-to-peer application and profile setting for second scenario

3.3.3 Third Scenario

In the third scenario, in additional to the three clients we have in the first two scenarios, client Peer3, Peer4, and Peer5 that connected to Router3, Router4, and Router5 respectively are added to the simulation. By doubling the amount of clients and kept packet size the same with the first scenario, to determine the effect on peer-to-peer gaming performance in terms of traffic and delays that are caused by number of peers. Below figure 3.3.3 shows the network peer-to-peer topology as we increased number of peers from 3 to 6, and figure 3.3.1b shows the application and profile setting that was used in third scenario simulation.

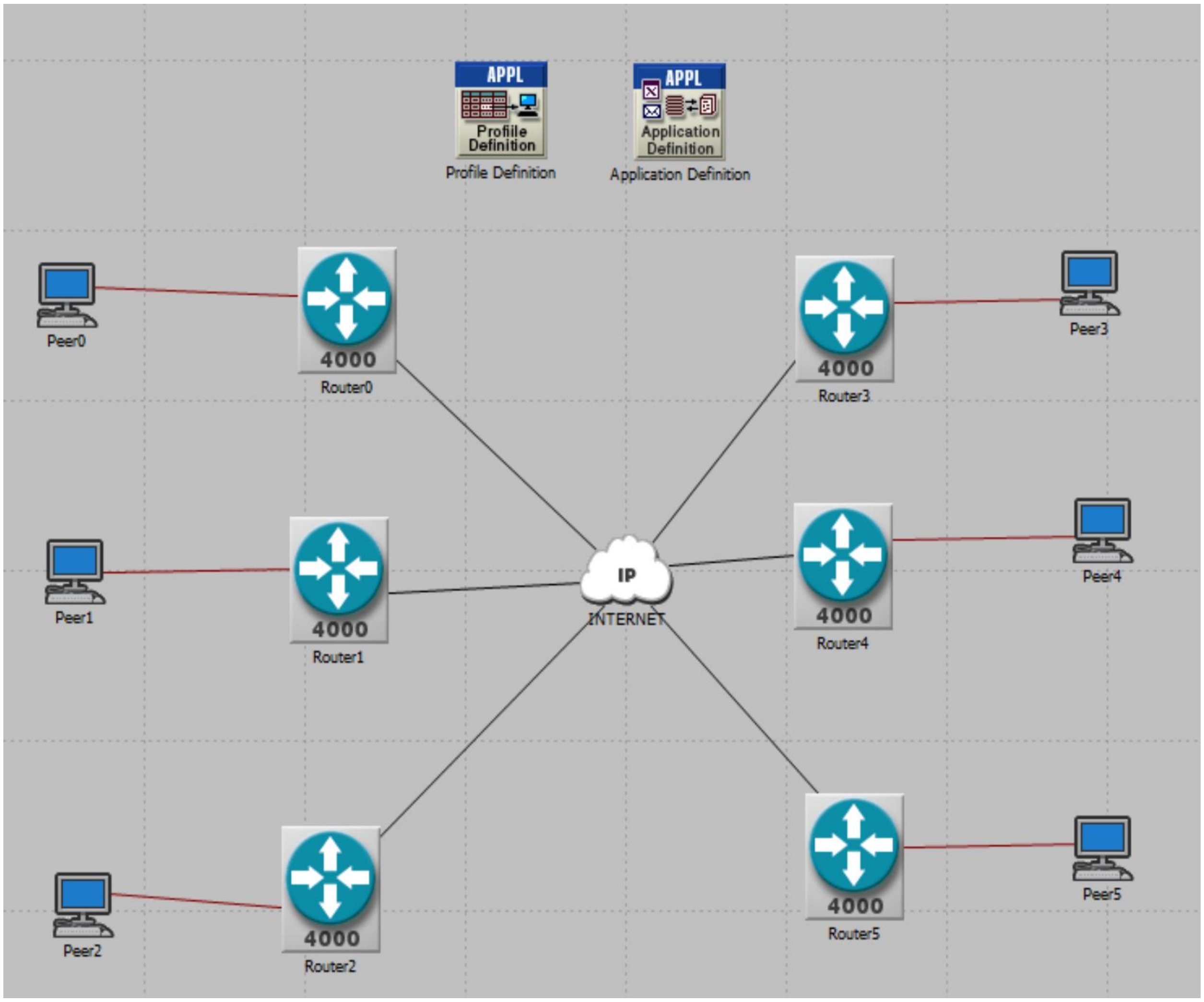


Figure 3.3.3: Network Topology for Peer-to-Peer Simulation in the Third Scenario

3.4 Simulation Results

Based on the setup and implementation described in the previous section, we simulated results from for all of the first, second and third scenarios. Different data are collected for each of the scenarios. Namely, Average Traffic Dropped (packet/sec.), Average End-to-End Delay (sec.), File Sharing Traffic Sent (packet/sec.) and File Sharing Download Response Time (packet/sec.) in order to produce meaningful results for analyzing the performance effects of various setups on P2P networks.

3.4.1 Traffic Dropped (Packets/Sec)

Analyzing the overall results of the average traffic dropped for all scenarios, we observe that there is an initial spike of dropped packets at the start of the simulation; from 0 minutes to around 5 minutes. Then, from 5 minutes to the end of simulation period, the average traffic dropped ratio showed an exponential decrease.

Observe the similarities between the first and second scenarios in Figure 3.4.1a and Figure 3.4.1b. In the 0 minutes to 5 minutes initial interval of those scenarios, the average traffic dropped ratio hovers between 0.25 to 0.30. Towards the end of the simulations the values eventually settled at between 0.00 to 0.05.

Although the results from the third scenario took on similar trend with the ones before it, it differs drastically in values. Examining Figure 3.4.1c, the initial 0 minutes to 5 minutes interval, the average traffic dropped ratio are between 0.50 and 0.60; doubled the values in the other two scenarios. Then, its higher rate of exponential decrease caused values at close to 40 minutes to settle at between 0.10 and 0.05.

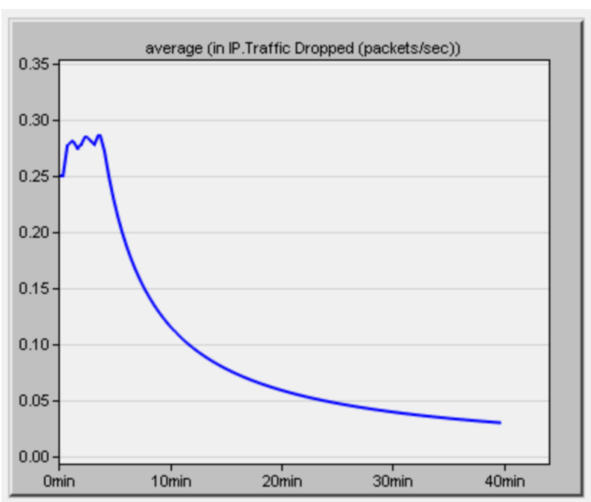


Figure 3.4.1a: First Scenario Average Traffic Dropped (packet/sec.)

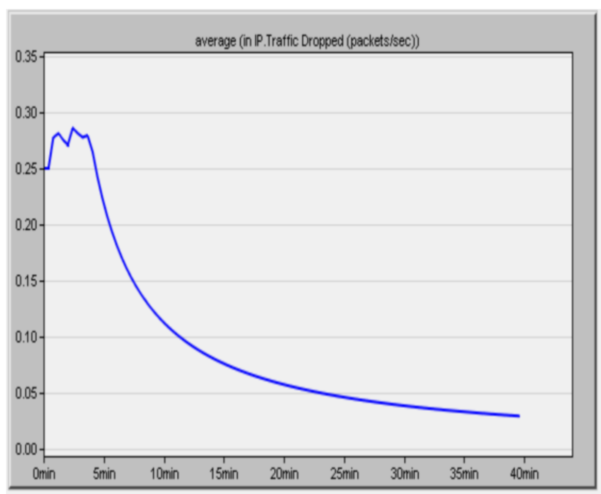


Figure 3.4.1b: Second Scenario Average Traffic Dropped (packet/sec.)

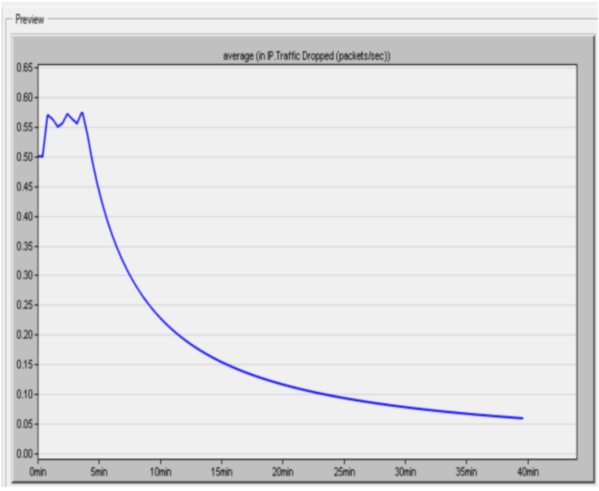


Figure 3.4.1c: Third Scenario Average Traffic Dropped (packet/sec.)

3.4.2 End-to-End Delay (Sec)

As seen from the figures below, where the average end to end delay is examined, by comparing the first scenario to the second scenario, we can see a dramatic increase on end to end delay when we increased our packet size from 80 bytes to 8000 bytes, which the maximum delay increased from around 0.00008 seconds in Figure 3.4.2a to around 0.00038 seconds in Figure 3.4.2b.

By comparing the first scenario to the third scenario, we can also observe a dramatic increase on end to end delay when we increase our peers from 3 clinents to 6 client, which the maximum delay increased from around 0.0008 seconds in Figure 3.4.2a to around 0.00021 seconds in Figure 3.4.2c.

We can conclude that both increasing packet size and number of client will have a dramatic effect on average end-to-end delay.

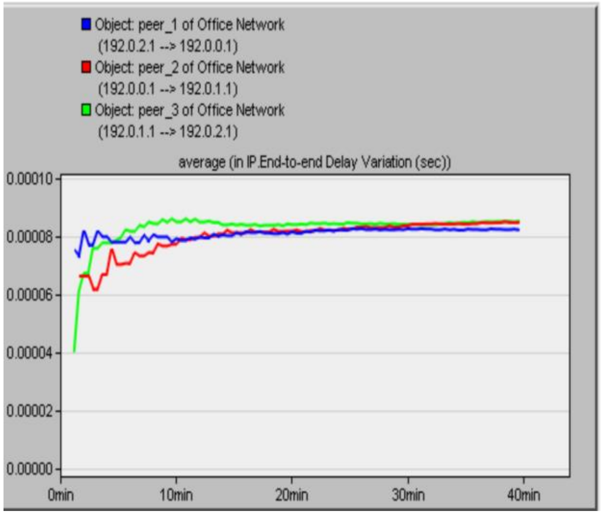


Figure 3.4.2a: First Scenario Average End-to-End Delay (sec.)

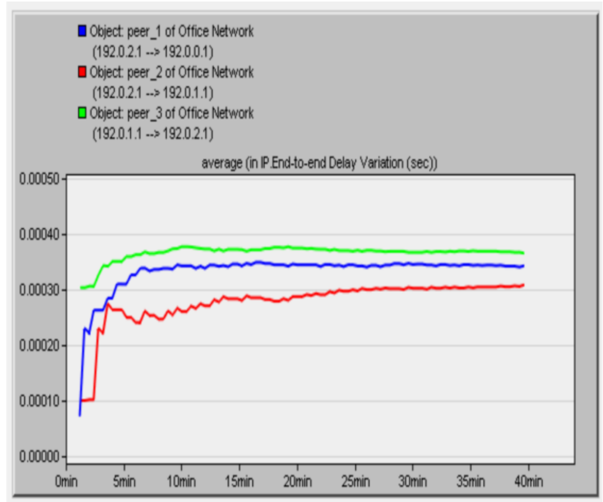


Figure 3.4.2b: Second Scenario Average End-to-End Delay (sec.)

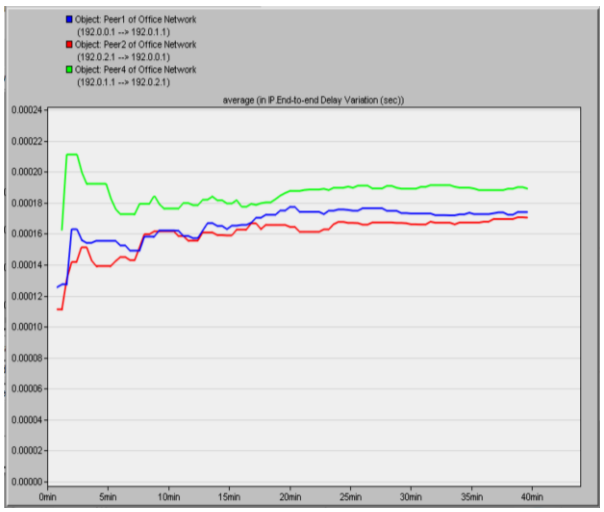


Figure 3.4.2c: Third Scenario Average End-to-End Delay (sec.)

3.4.3 Peer-to-Peer Traffic Sent (Packet/Sec)

As seen from the results of traffic sent in figures below, by comparing the first scenario and the second scenario. Such that as we increased the packet size from 80 bytes to 8000 bytes, we can observe a slight decrease in average traffic sent transmission rate, which the peak rate dropped from around 0.65 packets/second Figure 3.4.3a to 0.55 packet/second Figure 3.4.3c.

Furthermore, by comparing the first scenario to the third scenario, as we double amount of peers from 3 clients to 6 clients, we can also observe a slight decrease in traffic sent transmission rate, which the peak rate dropped from around 0.65 packets/second Figure 3.4.3a to 0.6 packets/second Figure 3.4.3c.

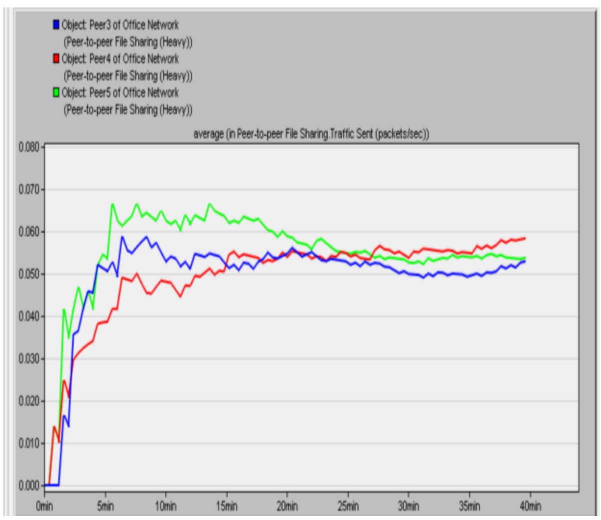


Figure 3.4.3a: First Scenario File Sharing Traffic Sent (packet/sec.)

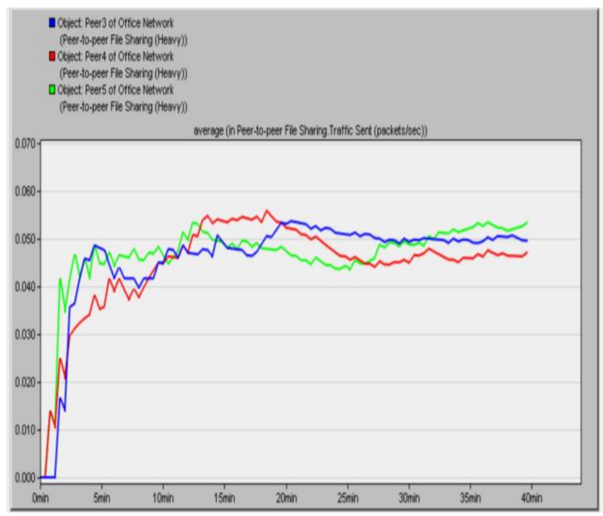


Figure 3.4.3b: Second Scenario File Sharing Traffic Sent (packet/sec.)

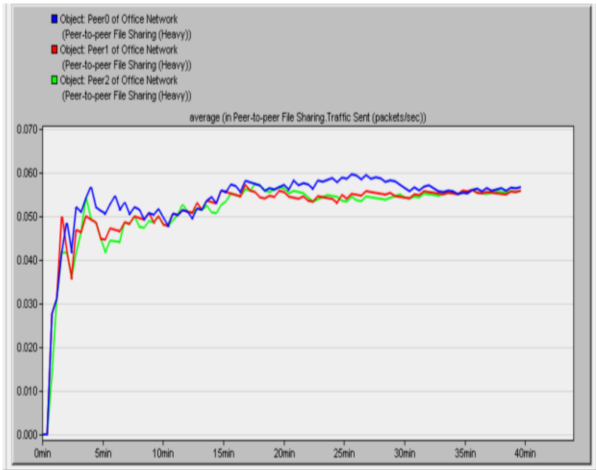
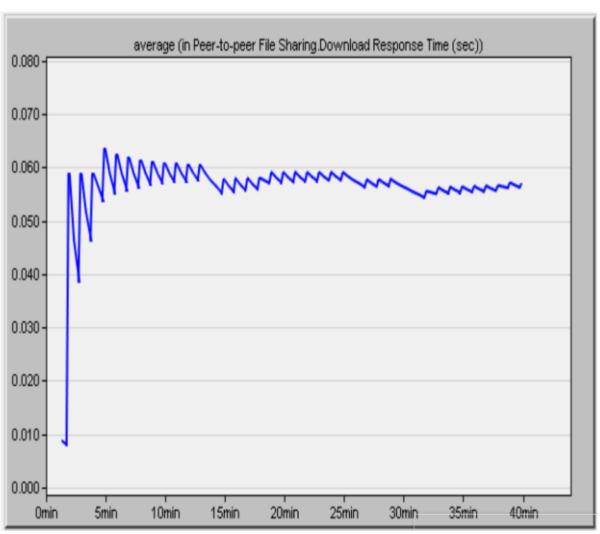


Figure 3.4.3c: Third Scenario File Sharing Traffic Sent (packet/sec.)

3.4.4 Peer-to-Peer Download Response Time (Sec)

The Download Response time corresponding to a full round trip, from the first time the packet is sent to the time the first packet download is received. Comparing the results of the P2P Download Response time between three different scenarios, we observe that increasing the packet size halved the download response time; the average download response time eventually stabled at 0.02 seconds. While increasing the number of client peers to the system, the average download response time doubled. This is expected result. The average download response time increased to 0.06 seconds and stays above 0.13 seconds.



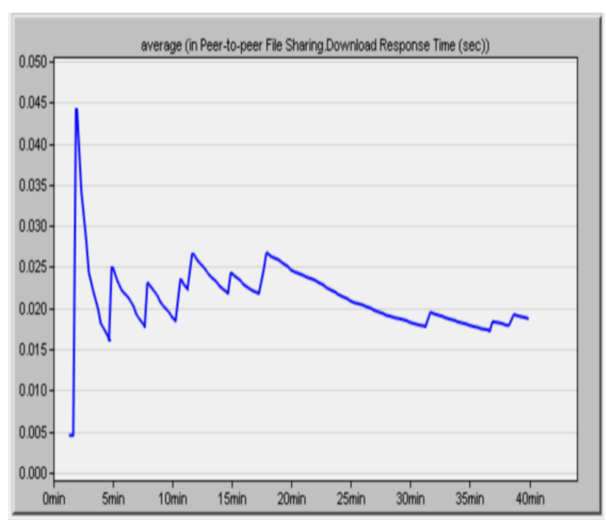


Figure 3.4.4a: First Scenario File Sharing Download Response Time (sec.)

Figure 3.4.4b: Second Scenario File Sharing Download Response Time (packet/sec.)

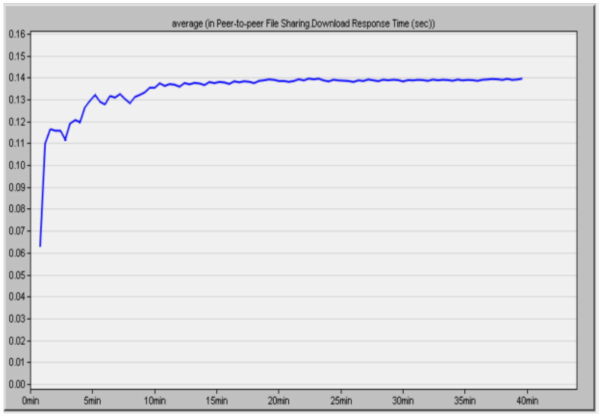


Figure 3.4.4c: Third Scenario File Sharing Download Response Time (packet/sec.)

4. DISCUSSION AND CONCLUSIONS

The P2P architecture was analyzed using the Riverbed Modeler by simulating three distinct scenarios. According to our peer-to-peer network simulation results, as the packets size is increased, its impact on the rate of packets dropped is negligible. However, the average end-to-end delay increased significantly and the traffic sent rate also saw a slight decrease. But the effect for download response time was positive as it was roughly halved. With a doubling of the number of peers in the system, the rate of packets dropped also doubled. We also observed significant increase on the end-to-end delay, as well as reduced traffic sent rate, and a doubling of the average download response time.

With 2 times more hosts joining to the system, the rate of packets dropped has doubled, it has a great increase on the end-to-end delay, the traffic sent rate has decreased slightly and average download response time is doubled. The difficulties of our project involving our project include setting up a device to device network, and simulate real world gaming aspect VIA P2P architecture, however, we are confident that these issues can be solved in the future.

In the future, we aim to compare different P2P systems such as Pure, Hybrid, and super-peer. We would also like to assess more characteristics for performance tradeoffs, and testing how varying the capabilities of the peers on the network affect the overall performance.

This project is non-trivial because the future for professional gaming has a great potential in terms large amount of reward funds, by increase stability and decrease latency, a fair battleground can be established. We all know the frustration when playing games with friends can be an awful experience due to the network latency, the fact that we do not need to relay on a center host and having to hope the server does not stop operate, we have eliminated some uncertainty and made our gaming experience more reliable and fun.

5. REFERENCES

[1]F. Bevilacqua, "Building a Peer-to-Peer Multiplayer Networked Game", *Game Development Envato Tuts+*, 2013. [Online]. Available: https://gamedevelopment.tutsplus.com/tutorials/building-a-peer-to-peer-multiplayer-networked-game--gamedev-10074. [Accessed: 08- Feb- 2019].

[2]C. GauthierDickey, D. Zappala, V. Lo and J. Marr, "Low Latency and Cheat-proof Event Ordering for Peer-to-Peer Games", *delivery.acm.org*, 2019. [Online]. Available: http://delivery.acm.org/10.1145/1010000/1005877/p134-gauthierdickey.pdf?ip=142.58.76.52&id=1005877&acc=ACTIVE%20SERVICE&key=FD0067F557510FFB%2E87ED2FCA0B39BEF5%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&__acm__=1549665331_278400722e186e645275845f2231b912. [Accessed: 08- Feb- 2019].

[3] C. Neumann, M. Varvello, N. Prigent and K. Suh, "Challenges in Peer-to-Peer Gaming"*Ccr.sigcomm.org*. [Online]. Available: http://ccr.sigcomm.org/online/files/p2p_gaming.pdf. [Accessed: 08- Feb- 2019].

[4]A. Yahyavi and B. Kemme, "Peer-to-peer architectures for massively multiplayer online games", 2013. [Online]. Available: https://www.contrib.andrew.cmu.edu/~ayahyavi/files/Yahyavi-CSUR13-P2PMMOG.pdf. [Accessed: 08- Feb- 2019].

[5]E. Buyukkaya, M. Abdallah and R. Cavagna, "VoroGame: A Hybrid P2P Architecture for Massively Multiplayer Games - IEEE Conference Publication", *Ieeexplore.ieee.org*, 2019. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4784788. [Accessed: 08- Feb- 2019].

[6]C. Neumann, M. Varvello, N. Prigent and K. Suh, "Challenges in Peer-to-Peer Gaming"Ccr.sigcomm.org. [Online]. Available: http://ccr.sigcomm.org/online/files/p2p_gaming.pdf. [Accessed: 25- March- 2019]

[7]Yang, B. and Garcia-Molina, H. (2019). Designing a Super-Peer Network. [online] Infolab.stanford.edu. Available at: http://infolab.stanford.edu/~byang/pubs/superpeer.pdf [Accessed 26- March- 2019]