Simulation of Massively Multiplayer Online Games Communication Using OPNET Custom Application

Sarmad A. Abdulazeez
Department of Computer Science
Faculty of Engineering and Technology
Liverpool John Moores University
Liverpool, UK
S.A.Abdulazeez@2013.ljmu.ac.uk

Abdennour El Rhalibi
Department of Computer Science
Faculty of Engineering and Technology
Liverpool John Moores University
Liverpool, UK
A.Elrhalibi@ljmu.ac.uk

Dhiya Al-Jumeily
Department of Computer Science
Faculty of Engineering and Technology
Liverpool John Moores University
Liverpool, UK
D.Aljumeily@ljmu.ac.uk

***Abstract*— In recent years, there has been an important growth in online gaming. Nowadays, Massively Multiplayer Online Games (MMOGs) may involve millions of synchronous players scattered around the world and engaging with each other within a single shared game. In this paper, we propose a new technique to communicate between players and game server, and between them based on hybrid Peer-to-Peer architecture. We propose to use OPNET Modeler 18.0, and in particular the custom application to simulate the new architecture, which required the implementation of new nodes models and behaviors in the simulator to emulate correctly the new architecture. We use OPNET Modeler 18.0 to simulate the network, applying two transport protocols TCP and UDP, and with different scenarios. The scenarios include both client-server and hybrid P2P system to evaluate the communication of games with (125, 500, and 1000) peers.**

Keywords-MMOGs; OPNET Modeler; TCP; UDP; hybrid Peer-to-Peer

I. INTRODUCTION

Online games generally involve a lot of actions that require fast response and low delay, and lost packet retransmission is impractical in these circumstances. Massively Multiplayer Online Games (MMOGs) are a type of online interactive game. As a result, traffic generated by online interactive games commonly consists of huge number of UDP packets. The traffic is also highly periodic because this requires consistency and updating between the players and the game server states. One of the key important features in these games is the massive number of players who play concurrently over the Internet. The huge number of players leads to a more interactive, complex, and attractive game environment. The main reason that makes MMOG's popularity increasing so rapidly in recent years is the experience of playing with other human players. With the increasing number of players in MMOGs, It became difficulty to control efficiently the communication between the players and the game server, and between players. There are historically two main architectures that used for MMOGs communication, although new cloud based architectures are being currently developed. The traditional architecture called client-server, in which a centralized server has the authority to perform the majority of the game logic and then send a game state update to the clients for rendering. The responsibility of client is to collect the user commands and send them to the server for processing. In this architecture, the requirement of computational power and bandwidth for a single client is minimal. However, considerable resources and consecrated support staff are required for a large number of players at the server side. In most case this can lead to a high delay, as well as high traffic of sent and received packets, which may consequently lead to low game performance. The other architecture called Peer-to-Peer architecture. P2P architecture corresponds to a network without a centralized server or control. Peer-to-peer networks allow game designers to transfer a big part of the game processing and load of communication bandwidth to their participant's peers. However, P2P architecture has some drawbacks to manage and control the system, due to there being no centralized control to guarantee each player in the game uses the same software, and does not engage in irregular game activities. In this paper, we use hybrid Peer-to-Peer architecture [1] for game communication between players and game server, which is implemented and deployed in OPNET Modeler 18.0 using custom application. Standard application in OPNET Modeler [2] does not support in game communication, therefore, we use custom application to deal with this aspect of the traffic and carry out evaluation of the different architecture using different scenarios. The rest of the paper is organized as follows. Section II presents a review of the state-of-the-art of MMOGs simulation systems. Section III describes the MMOGs communication in client-server system, while section IV introduces the MMOGs communication in P2P system and section V MMOGs communication in hybrid P2P system. Section VI illustrates the custom application in OPNET Modeler. Section VII, VIII, and IX introduce the simulation, results, and discussion. Section X presents the conclusion.

II. LITERATURE REVIEW

Game simulation is widely used evaluate new architectures and protocols. The game communication is considered one of the most important aspects in the game simulation. Some researches tried to deal with the simulation of massively multiplayer online game communication. Mohorko et al. present an overview of some suitable simulation tools for research on network technologies and communication infrastructure. The authors also show different possibilities of using the advanced

simulation frameworks for simulating the tactical networks. This system is of limited use for OPNET. This limitation leads to difficulties to control and change the communication model or create a new custom application [3]. Shirmohammadi et al. propose a P2P communication architecture for game network to provide reliability for important messages and decreasing network congestion by acknowledging only key messages. The system proposed called (HDSP) Hybrid Distributed Simulation Protocol. HDSP supports both best-effort delivery, for frequently occurring messages, and reliable delivery for important or "key" messages. HDSP simply implements the standard UDP protocol [4]. Denault and Kienzle used simulation in the context of MMOGs. They use Mammoth to simulate MMOGs to evaluate performance measurements such as CPU usage, memory usage. The authors define five simulation setups within the Mammoth MMOG framework, four using a single computer and the others in a distributed setting. The limitations of this system are the use of TCP protocol only, and there is no way to clearly define the game communication in the simulation [5]. Webb et al. develop an application layer Network Game Simulator (NGS) and successfully simulated Client/Server system, Region based, and Neighbor based architectures. The simulator is very flexible and can easily be extended to contain new features and architectures. The disadvantages of using this simulator are that there is no complex routing algorithms, and NGS does not provide support to simulate the network stack. In addition this simulator has been used to just perform preliminary evaluation [6]. All these researches do not use OPNET Modeler simulation to simulate the game communication or the custom game communication. The used of OPNET custom game communication [7] provide more flexible features to manage the communication and control the traffic flow between network devices in both architecture, and support both TCP and UDP protocols. The next section will introduce the MMOGs communication in client-server system.

III. MMOGs Communication in Client-Server Architecture

The vast majority of the MMOGs have been developed using the classic client-server architecture. This architecture demonstrates some problems such as all the load of handling the virtual world is carried by the server. The players connect to the server through the client application on their own computer and the server is responsible for handling all the rules and the state of the virtual world. One of the first things we have to be aware of is that due to the big number of clients that will connect to the server as well as the massive load that they create it will be impossible for one single server machine to handle everything. A lot of effort is put into developing better and more effective ways to deploy the load over multiple servers. This method can achieve the largest possible number of the participating players and experience with a minimum of needed resources.

One of the most common optimizations used by game communication is area of interest (AoI) management. Individual players usually only interact with a small area of the game world at any one time. Servers only need to update clients about objects in this AoI. This leads to reduce the number of objects transferred from the total set of objects to the number of objects in this area [8].

IV. MMOGs Communication in Peer to peer Architecture

The primary requirement for MMOGs based on Peer-to-Peer architecture is to maintain state consistency, and a shared sense of virtual space among great numbers of players without the need for support by the server [9]. The development of a peer to peer architecture for a MMOG is a significant research topic. There is a great deal of research which has been done on the use of peer to peer architectures to design MMOGs, but until now this has not led to it being used in any commercial MMOG. Because the use of peer-to-peer system to design and develop multiplayer games is a new phenomenon and it hasn't reached the mainstream consumer market. P2P architecture allows clients to communicate directly with each other and they are responsible for a small portion of the game state, and therefore do not need for a central game server. Dispensing the need for a central game server, you will decrease the cost of running a game dramatically. However, there are a lot of problems when designing and developing a peer-to-peer architecture for MMOGs. For example, a major problem will be to save the game state, the deployment of game updates will be more troublesome, and the bandwidth used for the clients will be extremely higher when compared with the client-server architecture. Peer-to-peer architectures do not have central server and therefore it becomes most difficult to present single consistent virtual world to all users. In Peer-to-Peer system, distributing the game state over all the peers in the game world, so there will have to be an effective way to set different elements of the game world to different peers [9].

V. MMOGs Communication in Hybrid Peer to Peer Architecture

Massively Multiplayer Online Games are becoming more widespread each year with diverse new technology being released on platforms. These developments make the game more exciting to attract additional numbers of players to the game. The increase in the number of players creates issues to be faced by gaming companies such as adding more servers to deal with this problem. However, this solution is not always possible because of the cost of adding new server is high, and the communication of state dissemination is also more complex when the players are managed from different servers. To better solve this problem, we propose a new technique based on Hybrid Peer-to-Peer system [1].

VI. CUSTOM APPLICATION IN OPNET MODELER

The custom application is a generic model framework that you can use to represent a broad class of applications. It can be used when the application needed in simulation does not correspond to any of the standard applications in OPNET. The standard applications in OPNET Modeler are for example FTP, E-mail, HTTP, Remote Login, Database, Video Conferencing, Print, Peer-to-Peer Sharing, Video Streaming, and Voice. The custom application provides attributes that allow you to configure different aspects of the application in detail. Custom application can be used to represent complex multi-tier applications. The following components are used within the custom application [10]:

- Task: A basic unit of user activity within the context of an application. For example: reading an e-mail message, obtaining records from a database or send a query to the server. The start and end of a task must be clearly defined.
- Phase: An interval of related activity that is contained in a task, for example a processing phase and a data transfer phase are specific phases of a task.
- Step: A phase consists of many steps, for example obtaining a simple HTTP browsing from a web server involves two steps: first, sending a request to the web server, and second, receiving the corresponding results from the web server.
- Tier: A single software process that used for executing some or all of the steps in a task.

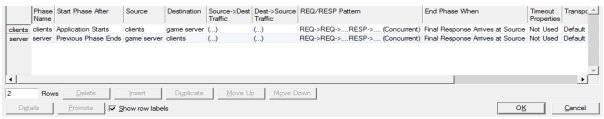

Fig. 1. Custom Task Manual Configuration

VII SIMULATION DESIGN

In this research, we have used OPNET Modeler 18.0 to implement and simulate the proposed architecture (MMOGs based on hybrid Peer-to-Peer system) and compared the results with the traditional architecture (Client-Server system). Furthermore, we compare transport protocol TCP versus UDP for each scenario in the project. The network topology consists of three scenarios for each architecture which will be explained in the following sections.

A. Network Topology for Client-Server system

This Client-Server architecture consists of the following devices: game server, two Cisco C400 Router, IP Cloud, Ethernet Switch, and Ethernet Workstations as shown in figure 2. The connection mechanism between these devices is described as follow: The game server connects with the IP Cloud through Cisco C400 Router (server gateway). The IP cloud connects with the Ethernet Workstations (Peers) through both Cisco C400 Router (Client gateway) and Ethernet Switch. We use Ethernet 100BaseT link to connect between server and server gateway, also between the client gateway and Ethernet switch. We use PPP DS1 link to connect between Router and IP Cloud, also between IP Cloud and Client gateway. We use Ethernet 10BaseT link to connect between the Ethernet Workstations and Ethernet Switch. The number of Ethernet Workstations is (125, 500, and 1000) for each scenario. The game applications are controlled and managed by the central server. We use task configuration to configure the custom application for this scenario.

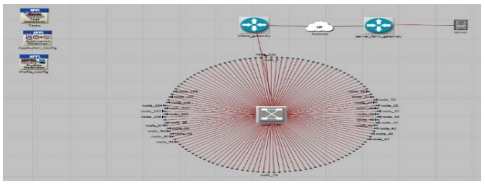

Fig. 2. Client/Server scenario

B. Network Topology for Hybrid P2P system

The hybrid P2P scenario has the same devices as in scenario 1 but with change in the zone area by adding super-peer and clone-super-peer to the region. Each region has 60 peers connected to both super-peer and clone-super-peer using the super-peer gateway router as shown in figure 3. The game applications is controlled and managed by the region super-peer and clone-super-peer. We use task configuration to manually configure the custom game application for hybrid P2P system.

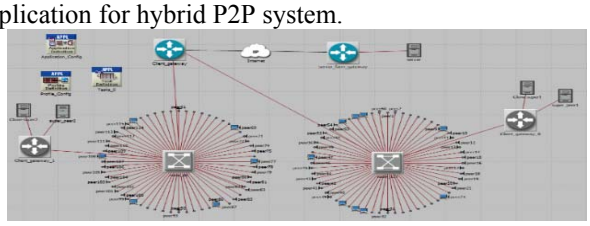

Fig. 3. Hybrid P2P scenario

C. *Custom Application for Game Communication*

As the hybrid P2P architecture is not currently supported in OPNET, or any other simulator, we have created a custom application for game communication between client and server in client-server architecture, and between peers, super-peers, clone-super-peers, and server in hybrid P2P architecture. The custom application for client-server architecture consists of two phases. The first phase is to communicate between clients and server, however the second phase is to communicate between server and clients. However, the custom application for hybrid P2P architecture consists of 20 phases for the scenario with 125 peers, however the number of phases will increase whenever the number of peers is increasing till up to 80 phases for the scenario with 1000 peers and 16 regions. The number of regions will increase with the increase the number of peers. Figures 4 and 5 show the custom application for client-server architecture. In figures 4 and 5, the phase 1 is used to communicate clients with the server, however the phase 2 is used to communicate the server with clients. Figure 6 and 7

show the custom application phases for hybrid P2P system that is used to communicate the peers with both, regions super-peer and clone-super-peer, as well as the communication between server, super-peer and clone-super-peer. The OPNET custom application provides some advantages for hybrid P2P architecture. These advantages are game communication, boot strapping, management of the login and leaving of player, control the migration from one region to another, easy to manage the increase of players, and manage the failure and recovery.

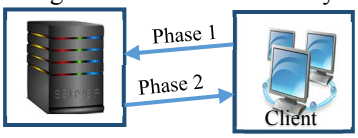

Fig. 4. Custom Application phases in Client-Server System

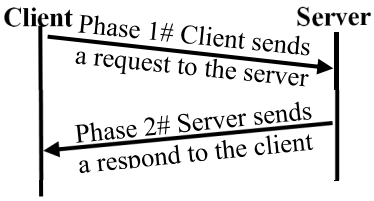

Fig. 5. Phases Description in Client-Server System

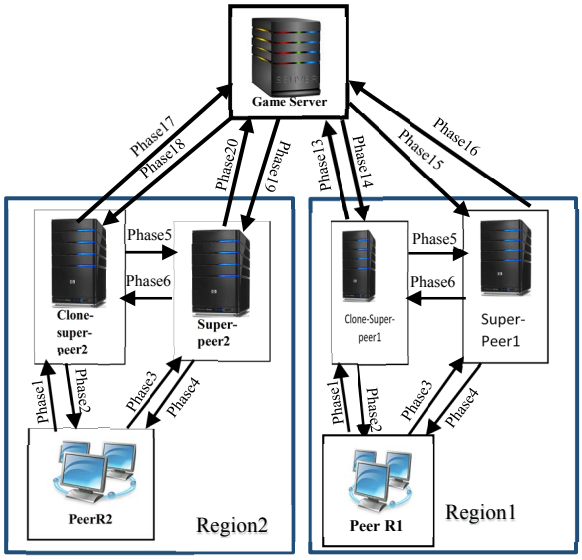

Fig. 6. Custom Application phases in Hybrid P2P System

VIII. Simulation Results

With the scenarios that have been introduced previously, we have got to the following simulation results:

A. *Overall delay*

This parameter is defined as the overall end-to-end delay for all packets received by the station. End- to-end delay for the application used during the simulation is measured from the time from source to destination. End-to-end delay refers to the time it takes to send a packet from source to destination over a network. Figure 8 shows the overall delay for client-server architecture compared to the results of the overall delay for the hybrid P2P architecture using TCP transport protocol. The figure below illustrates a great variation of delay when using client-server system for game communication compared with hybrid P2P system. The client-server delay curve increases during the simulation time; however the curve of hybrid P2P is more stable after two second of the simulation time. In addition, figure 8 shows the effect of adding background traffic on base links of the network. However, figure 9 shows the overall delay for the same scenario but using UDP transport protocol instead of TCP. Further, the figure illustrates the impact of adding background traffic on base links of the network. Figure 9 illustrates that using UDP transport protocol is more stable for both architectures, as well as produce low delay.

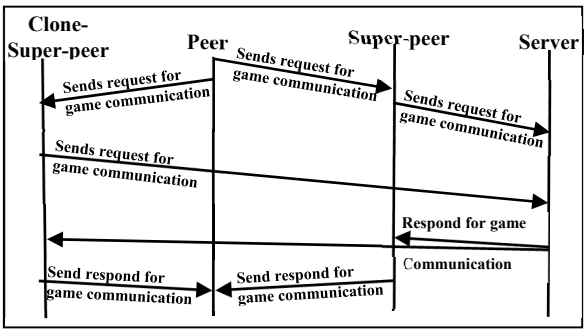

Fig. 7. Phases Description in Hybrid P2P System

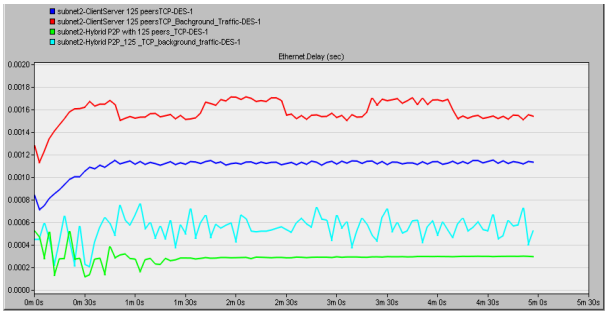

Fig. 8. Overall Delay for Client/Server and Hybrid P2P with 125 peers TCP Protocol with Background Traffic

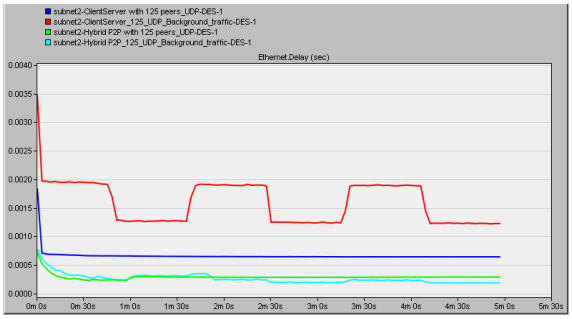

Fig. 9. Overall Delay for Client/Server and Hybrid P2P with 125 peers UDP Protocol with Background Traffic

Figure 10 shows the overall delay for both Client/Server and Hybrid P2P architecture with 500 peers using TCP transport protocol and the effect of adding background traffic on base links of the network. The figure below elucidates the using of hybrid P2P system produces less delay compare with client-server system. However, figure 11 shows the overall delay for both Client/Server and Hybrid P2P architecture with 1000 peers using TCP transport protocol without adding background traffic on base links of the network.

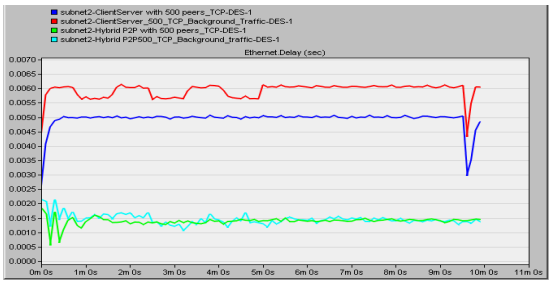

Fig. 10. Overall Delay for Client/Server and Hybrid P2P with 500 peers TCP Protocol with Background Traffic

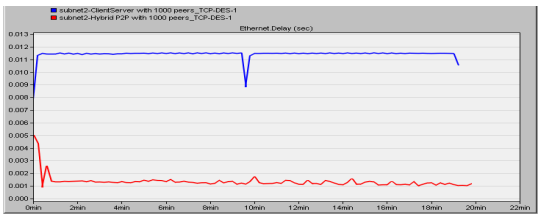

Figure 11. The overall Delay for Client/Server and Hybrid P2P with 1000 peers TCP Protocol without Background Traffic

B. *Traffic Received*

Traffic received is the average number of bytes per second received by all the nodes in the network. In other words, network traffic is the amount of data moving across a network at a certain point of time. Network traffic is one of the main ingredient for measuring network traffic, network traffic control and simulation. The proper regulation for network traffic helps to ensure the quality of service for the network. Figure 12 and 13 show the custom application traffic received for client server and hybrid P2P system with 125 peers using TCP and UDP transport protocol respectively, as well as the effect of adding background traffic on base links of the network. The figures illustrate that the traffic received for client-server architecture is extremely large compared with the traffic received for hybrid P2P architecture. As well as the client-server traffic received continue increasing till the end of the simulation, however the hybrid P2P traffic received was more stable with a little bit increasing after half of simulation time.

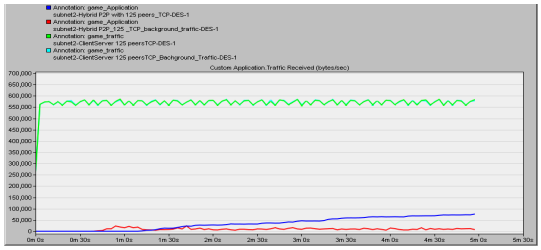

Fig. 12. Traffic Received for Client/Server and Hybrid P2P with 125 peers TCP Protocol with Background Traffic

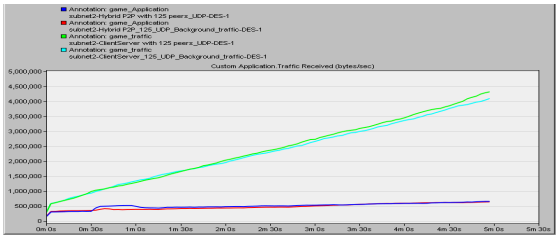

Fig. 13. Traffic Received for Client/Server and Hybrid P2P with 125 peers UDP Protocol with Background Traffic

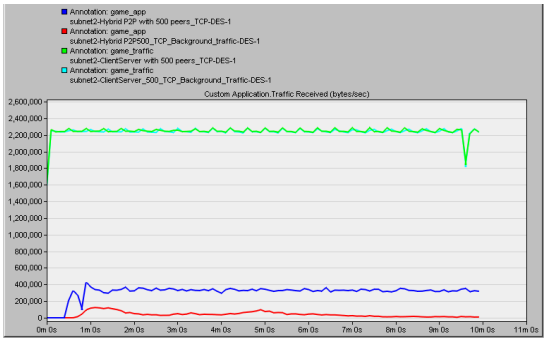

Fig. 14. Traffic Received for Client/Server and Hybrid P2P with 500 peers TCP Protocol with Background Traffic

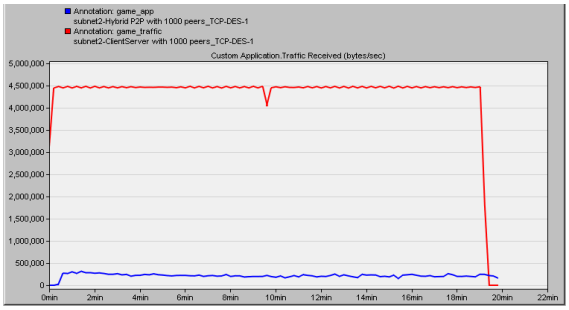

Fig. 15. Traffic Received for Client/Server and Hybrid P2P with 1000 peers TCP Protocol without Background Traffic

Figures 14 and 15 illustrate the traffic received for both client-server and hybrid P2P architectures with 500 and 1000 peers using TCP transport protocol. The figures above show that the traffic received by client server architecture is much higher than the traffic received by hybrid P2P architecture for both scenarios. The traffic received for client server is duplicated when the number of clients is duplicated from 500 to 1000 clients, however, the traffic

received for hybrid P2P is slightly higher with 1000 peers. Due to the using of hybrid P2P architecture is more organized and more efficient compare with client-server architecture.

C. *Ethernet Throughput*

Throughput represents the average number of bits successfully received or transmitted by the receiver or transmitter channel per unit time, in bits per second. It allowed us to monitor the average rate of successful message delivered over a channel. Figures 15 and 16 show the throughput of edge client in client-server architecture compare with the throughput of peer in hybrid P2P architecture using TCP protocol without add any background traffic to the base link. The figures illustrate that the client-server architecture produce more throughput compare with hybrid P2P architecture. As well as, the throughput of client-server system is increasing, while the throughput of hybrid P2P system is more stable.

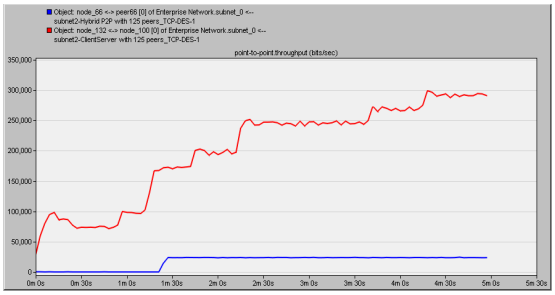

Fig. 15. Throughput out for Edge Client and Peer

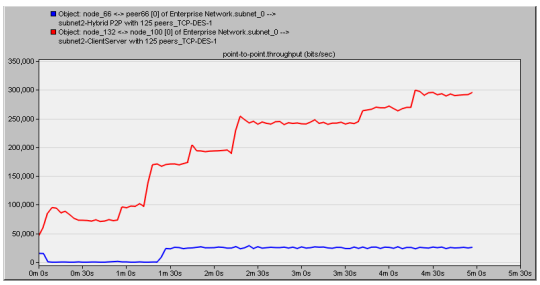

Fig. 16. Throughput In for Edge Client and Peer

IX. DISCUSSION

Typically, network simulation is designed to reproduce a system as close as possible and accurately from real world conditions. The experimental results presented in section VIII demonstrate that the simulated delay, traffic received, and throughput for the hybrid P2P architecture is low compare with the client-server architecture. In addition, the experimental results for custom application of OPNET is more realistic than the experimental results for standard application [11].

X. CONCLUSIONS

In this paper, we have reviewed the research related to the MMOGs communication and simulation, as well as explained the advantages and limitation of these researches. We also highlight the subjects and issues related to our research and explain the main contributions of the paper. We use OPNET Modeler 18.0 to model and simulate the communication networks. We have used OPNET simulation to enable the networks construction, study of communication infrastructure, design of individual devices, and simulation of protocols and applications. The results illustrate that the hybrid Peer-to-Peer system produce low delay, low traffic received, and low throughput in game communication when compared with client-server system for all scenarios used in the simulation.

REFERENCES

[1] S. A. Abdulazeez, A. El Rhalibi, M. Merabti and D. Al-Jumeily " Survey of Solutions for Peer-to-Peer MMOGs", 11th IEEE NIME collocated with IEEE ICNC 2015, Anaheim, California, USA.

[2] S. A. Sethi and V. Y. Huatyshin, *The Practical OPNET® User Guide for Computer Network Simulation*. CRC Press, Taylor & Francis Group, 2013.

[3] J. Mohorko, F. Matjaž, and K. Saša, "Advanced modelling and simulation methods for communication networks," *Microw. Rev.*, pp. 41–46, 2008.

[4] S. Shirmohammadi, A. Diabi, and P. Lacombe, "A Peer-to-Peer Communication Architecture for Networked Games," Int. Journal of Advanced Media and Communication, v.4 n.2, 2005.

[5] A. Denault and J. Kienzle, "The perils of using simulations to evaluate Massively Multiplayer Online Game performance," *3rd Int. ICST Conf. Simul. Tools Tech. ICST (Institute Comput. Sci. Soc. Telecommun. Eng.*, 2010.

[6] S. D. Webb, W. Lauel, and S. Soh, "NGS : An Application Layer Network Game Simulator," *Proceeding IE '06 Proc. 3rd Australas. Conf. Interact. Entertain.*, pp. 15–22, 2006.

[7] "Applications Model User Guide," in *Guide Standard Models, Modeler Relase 10.0*, pp. 1–36.

[8] B. Hughes, J. Haggerty, J. Nothman, S. Manickam, and J. R. Curran, "A Distributed Architecture for Interactive Parse Annotation," *Proc. Australas. Lang. Technol. Work.*, pp. 207–214, 2005.

[9] L. Fan, "Design issues for Peer-to-Peer Massively Multiplayer Online Games," *Int. J. Adv. Media Commun.*, vol. 4, no. 2, pp. 108–125, 2010.

[10] Z. Lu and H. Yang, *Unlocking the Power of OPNET Modeler*, Cambridge University Press, New York, 2012.

[11] 161. Sarmad A. Abdulazeez, Abdennour El Rhalibi and Dhiya Al-Jumeily "Evaluation of Scalability and Communication in MMOGs", 13th IEEE CCNC 2016 (IEEE Consumer Communications and Networking Conference), 9-12 January 2016, Las Vegas, Nevada, USA..