Project 3 - Client/Server Architecture: Multi-Tiered

Travis Wentz and Batuhan Mekiker

I. INTRODUCTION

In our previous client/server architectural experiment, we modeled a simple two-tiered architecture. We found that in having just one server to communicate with all our IoT devices and all our mobile clients, the single server quickly became overwhelmed and the network saturated with traffic. In fact, we found that with just 600 IoT devices which each communicate with the server every 30 to 120 minutes, the network quickly became saturated.

In this experiment, rather than a two-tiered architecture, we will examine the benefits of a multi-tiered architecture. Although the multi-tiered architecture proves more complex than a two-tiered architecture (due in large part to the need for summarization and intra-server communication) and incurs additional costs in terms of infrastructure, the multi-tiered architecture proves to be a better solution than the two-tiered architecture when the number of connected devices is higher than the 600 or so (with our experimental parameters) that the two-tiered architecture can handle. Our goal in this experiment is to quantify just how much more communication (and in our experiments, how many more IoT devices) the multi-tiered architecture can handle, compared to a two-tiered architecture.

II. RELATED WORK

According to our research, there could be couple of different network architectures, that could be employed to serve in different application. What we were interested in was different client/server infrastructures. One of the early examples of client/server architecture research is [1]. In this paper, Hanson, argues and compares couple of different client/server architectures and explores different possibilities and tries to find optimal solution for different problems. On the other hand, [3] and [2] argues for what would be the best solution for online cloud gaming environment for MMOG. They explore the performance of different network infrastructures and investigate the performance requirements for online gaming where [2] settles for peer-to-peer architecture. In recent years [4] has been published. In this paper,network topology is more like ours now. They updated the overall research in this area and compare the different architectures and evaluate them based on some parameters.In [5] due to high complexity nature of Augmented Reality (AR) and requirement for high bandwidth, existing network architecture can't actually support AR with high performance. In this paper, Westphal, argues what could be done to achieve higher performance with current technology.

The next three reference, explores the possibility of smart parking where there are hundreds of IoT devices that are connected to a network topology.The reason we added those are to give an idea about where we could use our design to achieve higher performance. In those, they mostly focus on application layer. Nevertheless, they give a brief detail how a smart parking system can shape a network infrastructure.

III. ARCHITECTURE OVERVIEW

The need for a multi-tiered architecture can be seen in many applications, but one application in particular is in the world of Massive Multiplayer Online Games (MMOGs). In MMOGs, it is nearly always impractical to connect every user to a single server. The amount of load on that single server is simply too much for that server to handle. Therefore, a multi-tiered architecture is generally used. The most common architecture is a four-layer architecture, wherein the four layers are: client layer, proxy layer, application/game layer, and database layer [3]. By splitting some necessary processes to different layers, the load that would otherwise be placed on a single server is reduced.

In another application which finds free parking spaces for a user (like the example used in our previous two experiments), if implemented in a large city such as L.A. a single server would likely not be able to handle the load. Furthermore, in a large city a single server would likely not be a logical solution. In such an application, there would be no need for all parking information for the entire city to be stored on a single server as no single user would need to see the exact location of every available parking space in L.A. at once. A better solution might be a multi-tier architecture where the lowest tier server may contain exact information about a single area and the levels above that server contain summaries about the information stored on the servers below them. From farther away, it would likely be enough to know that there is parking available in one particular neighborhood, but know nothing more specific. Then, as you get into that neighborhood, you could see exact information, from local servers, about which exact spaces are available.

One important note on multi-tiered architectures, is that with the wide-spread adoption of cloud computing, such an architecture has become much easier to implement and much more common [4]. This makes this solution highly scalable and as it is becoming cheaper, the technology is likely to be advanced further in the future.

IV. SIMULATION

This simulation was designed as follows: we had a variable number of IoT devices connected to their server, sending UDP messages every 30-120 minutes, which could represent

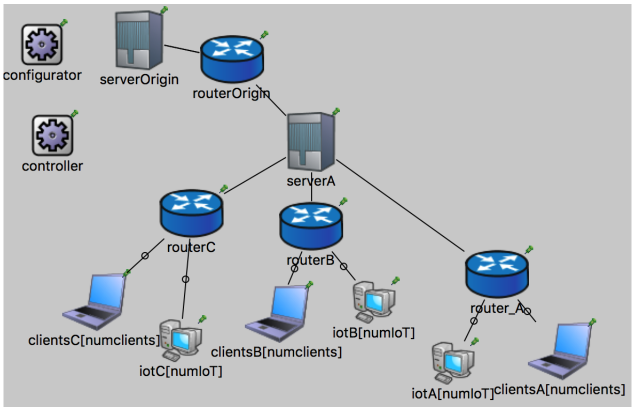


Fig. 1: Multi-Tiered Network Overview

parking spaces updating their status. We used OMNET's pre-configured modules such as *httptools* and network layers like *IPv4* to simulate our multi-tiered architecture just as we simulated our two-tiered architecture. In addition, we also simulated mobile clients contacting the server in random intervals from one half a second to one second via TCP, which could represent people trying to find an available parking space.

This architectural model, in contrast to a two-tiered architecture, complicates things some what, in that information gathered by a leaf-node server must be summarized and passed to its parent server. We chose to have the information passed to the parent server in small, UDP packets once per minute. The reason we chose once per minute was rather arbitrary. We decided that this interval might be a reasonable amount of time to update where available parking spaces are in a busy city. Additionally, there is further TCP communication between leaf-node servers and parent servers as mobile clients are not always inquiring about available parking spaces only in the area served by the closest server. We assumed that people may wish to know about available parking in a completely different area and can therefor, access summary information about the area in question. In this case, the mobile client's request must traverse the tree as far as needed to find a server with the summary of the area in question.

Our original expectation was that the bottleneck of the multi-tiered architecture would be much wider than the bottleneck of the two-tiered architecture. We expected that the multi-tiered architecture would be able to handle much more traffic than the two-tiered architecture.

V. RESULTS

As mentioned before, we chose to simulate our infrastructure on OMNET++. The simulation mostly focused on end-to-end delay, data rate measurements with varying number of IoT devices, and therefore load, and passed packet's size between the origin server (our highest level server) and server A (our lowest level server).

Results were largely as expected. As can be seen In Fig. 3, with increasing number of IoT devices data rate is decreasing, since the load on the network is larger. When the number of IoT devices is 30, the data rate is around 105 bits/sec. This number changes roughly to 95 bits/sec when the number of IoT devices increases to 90. The data rate reaches its lowest when the number of IoT devices is highest, which is at 600 IoT devices. In this case we see the data rate drops to 80 bits/sec.

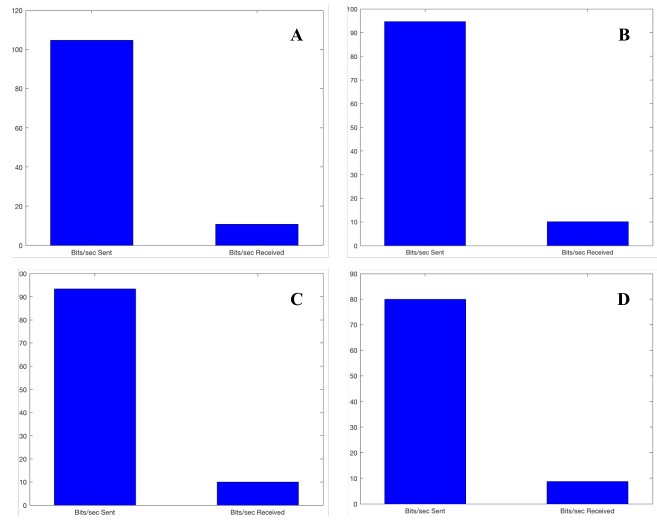


Fig. 2: Bandwidth Measurements in Bits/sec A-30, B-90, C-300 and D-600 IoT Devices

As previously stated, another metric that we were interested in was end-to-end delay. Fig. 4 in the appendix shows our results for end-to-end delay which was surprisingly acceptable and never passed the limit of 0.35 seconds. This was also the case in our two-tiered simulations. With the varying load on the network, the measurement became more granular and represented itself in a fixed fashion.

The last metric that we analyzed was the affect of packet size on the network. Figures 5a, 5b, 6a, 6b, 7b, and 7c in the appendix, all show the affect of varying packet sizes on the network for an increasing number of IoT devices. What we found was that the network quickly became saturated as soon as packet size was between 0.01 and 0.012 bits, regardless of the number of IoT devices creating traffic. This suggests to us that a packet size under 0.01 bits is interesting as under this threshold, the number of IoT devices begin to affect the network differently.

What our data show is that under the 0.01 bit packet size, the network saturation for server A sending communication to the origin server becomes unacceptable with 350x3 IoT devices (figure 7c). This is to be expected as the rest of our data show an increase in network saturation up to the point of 350x3 IoT devices.

At this point, we once again reached limitations based on our hardware. With just over 1000 IoT devices, each simulation took several hours to build and run. This gave us over a million data points to be sorted and interpreted. As expected, the trend we saw with a decreasing bit rate continued and, with 350x3 IoT devices, we discovered a bit rate of approximately 70 bits/second, which was the lowest point we reached. At this point, origin server started to become overwhelmed, yet remained fully operational.

In contrast to our two-tiered simulation, we were unable to reach a point at which our simulations failed all together. Due to hardware constraints which required an inordinate amount of time to build simulations this large, we were satisfied in deeming the network saturation too great when the end-to-end delay reached 70 bits/second.

What we did not expect, however, was that the multi-tiered architecture was unable to handle significantly more IoT devices than the two-tiered architecture. With the increased network traffic due to constant summarization updates, the network quickly became saturated with traffic. We had speculated that by using small, UDP packets for updates, rather than having a TCP session, the traffic would be small enough to not make a significant difference; however, our results were unexpected.

VI. FUTURE WORK

As future work, the first thing we would plan to do is alter the parameters of our simulations. Clearly, with leaf-node servers updating the origin server every minute, the multi-tiered simulation did not perform significantly better than our previously simulated two-tiered architecture. These parameters were chosen for this particular application, however, for a different application, the parameters may be altered such that a multi-tiered architecture would perform significantly better than a two-tiered architecture.

Next, we would plan to expand our infrastructure from three tier to more tiers and explore the possibilities. We would also plan to find optimal number of servers and clients to achieve higher performance.

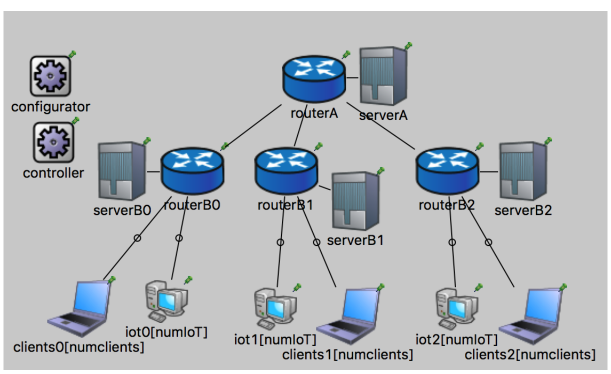


Fig. 3: Future Work

Since the main idea of multi-tier network is to share the load as much as possible. Fig. 3 illustrates possible network infrastructure that could exploit the benefits of multi-tiered architecture more efficiently.

VII. CONCLUSIONS

This paper explores some of the benefits and limitations of a multi-tiered architecture. Previously, we had explored a two-tiered architecture and it was our hope that by comparing the two architectures, a quantifiable line at which it became beneficial to use one architecture over the other would be found. By running simulations in OMNET++ we found that, in our multi-tiered simulations, the network became saturated with traffic when around 1000 IoT devices were connected. This was a surprising result, in that it was not so much better than a two-tiered simulation as we expected. However, we do expect that for a different application with slightly altered parameters, the multi-tiered architecture would perform significantly better than a two-tiered architecture.

REFERENCES

[1] M. David Hanson. "The Client/Server Architecture". Best Practices: Server Managment (2000). Chapter 1.

[2] Rhalibi A. E., & Merabti M., (2005). "Agents-Based Modeling for a Peer-to-Peer MMOG Architecture", Moores University, Liverpool, U.K., ACM Computers in Entertainment, Vol. 3, No. 2, April 2005, Article 3B.

[3] Moraal, Martijn. "Massive multiplayer online game architectures." Whitepaper, January (2006).

[4] Shea, Ryan, et al. "Cloud gaming: architecture and performance." IEEE Network 27.4 (2013): 16-21.

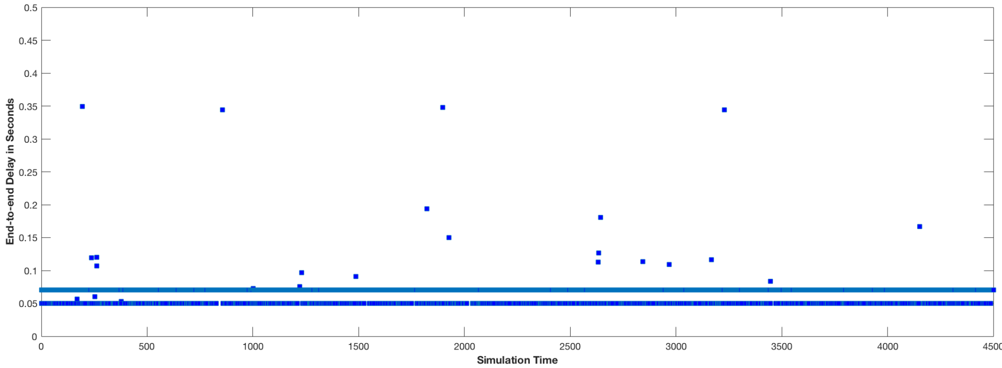
[5] Westphal C., Challenges in Networking to Support Augmented Reality and Virtual Reality (2017).

[6] Khanna, A. & Anand, R.,(2016). "IoT Based Smart Parking System", University of Petroleum and Energy Studies, IEEE.

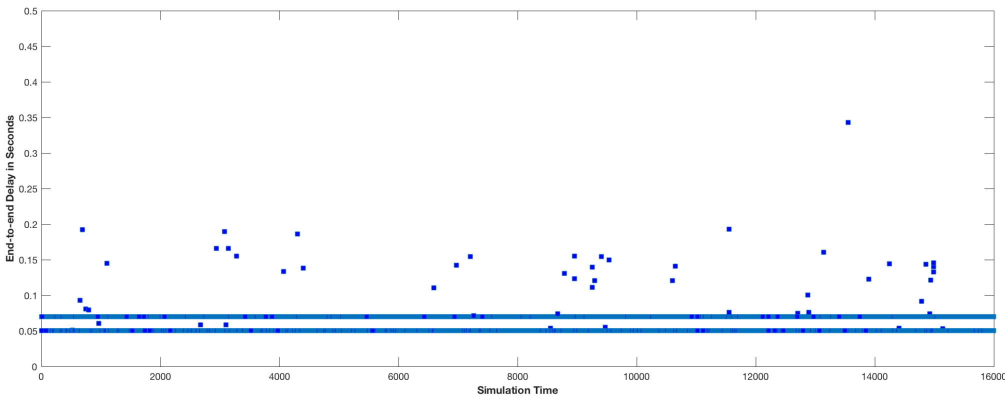
[7] Pham, T. N. et al.,"A Cloud-Based Smart-Parking System Based on Internet-of-Things Technologies", Feng Chia University & National Changhua University of Education, Taiwan. August, 2015.

[8] Adki P. R. et al., "Cloud assisted time-efficient vehicle parking services", International Conference on Inventive Computation Technologies (ICICT), IEEE, August, 2016.

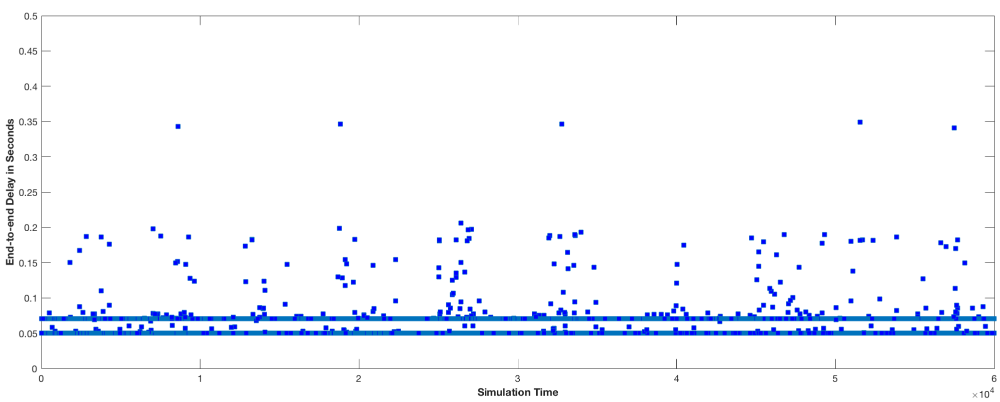
APPENDIX



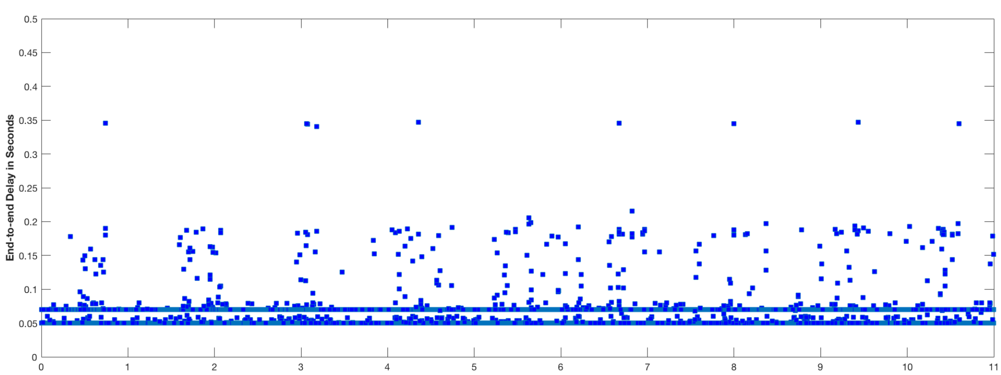
(a) End-to-end delay vs. simulation time. Number of IoT devices = 10x3



(b) End-to-end delay vs. simulation time. Number of IoT devices = 30x3

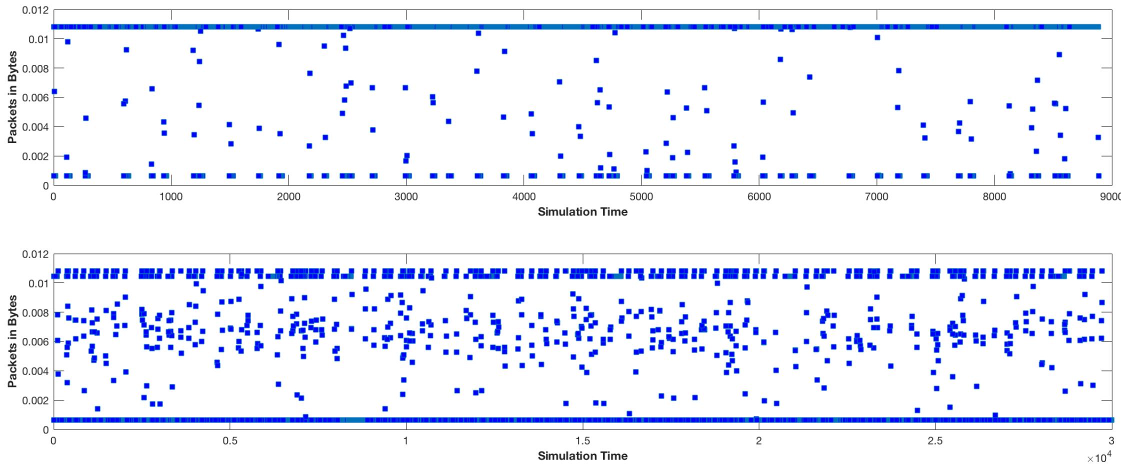


(c) End-to-end delay vs. simulation time. Number of IoT devices = 100x3

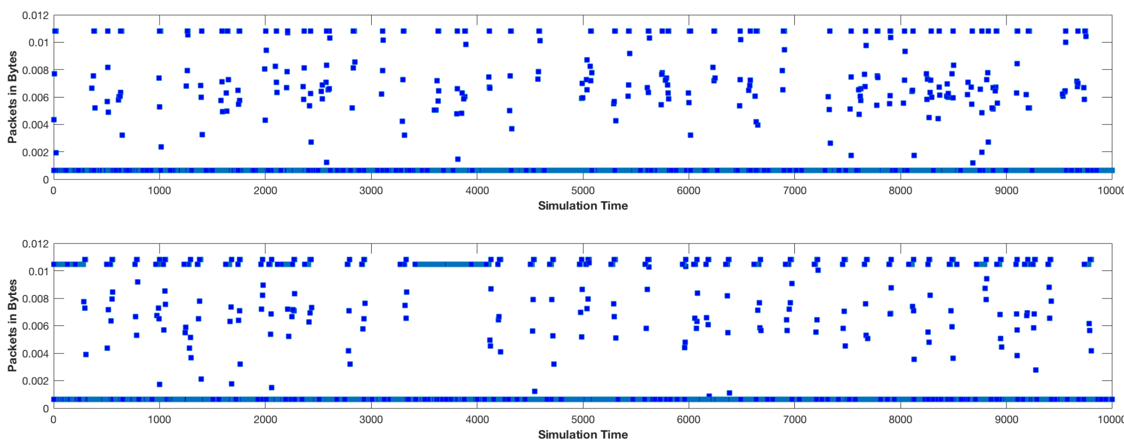


(d) End-to-end delay vs. simulation time. Number of IoT devices = 200x3

Fig. 4

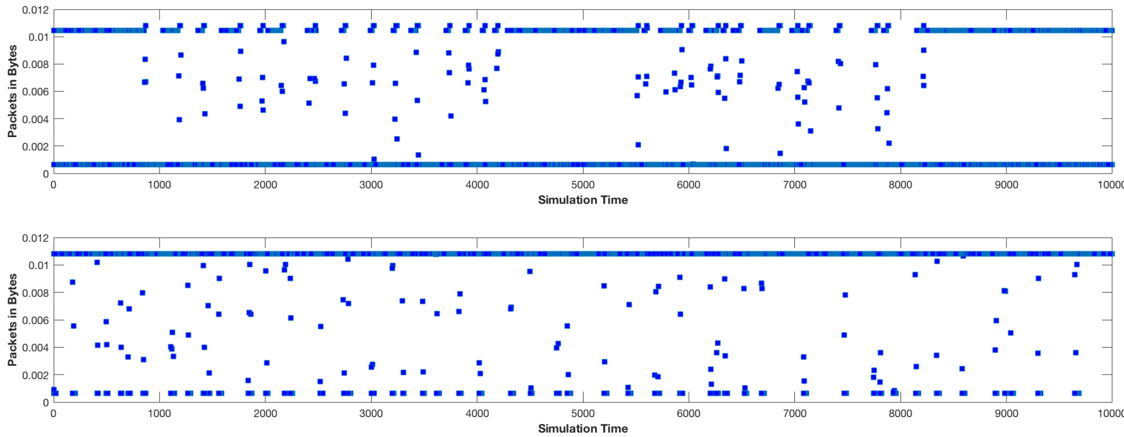


(a) Passed Up Packets from Server A to Origin Server vs. Origin Server to Server A, Number of IoT = 10x3

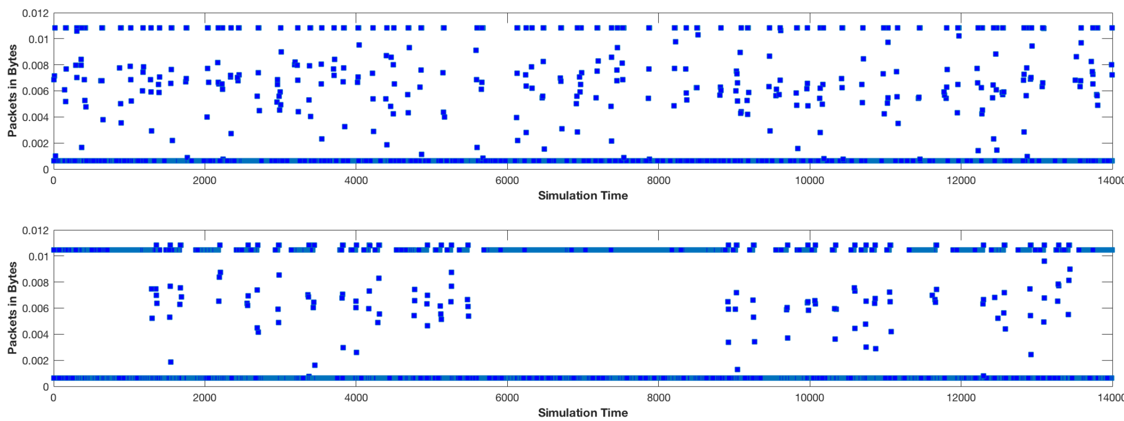


(b) Passed Up Packets from Origin Server to Server A vs. Server A to Origin Server, Number of IoT = 30x3

Fig. 5

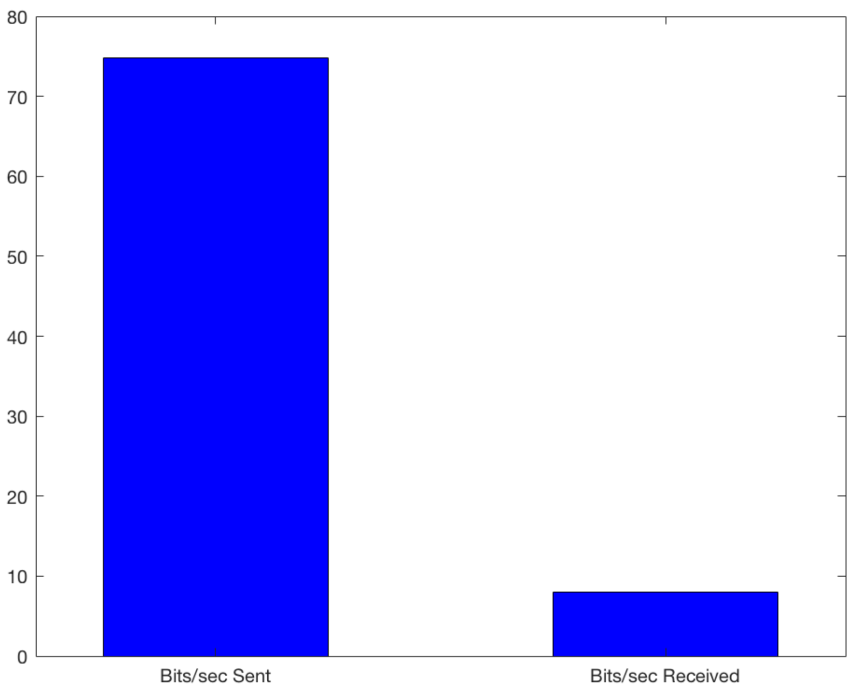


(a) Passed Up Packets from Origin Server to Server A vs. Server A to Origin Server, Number of IoT = 100x3

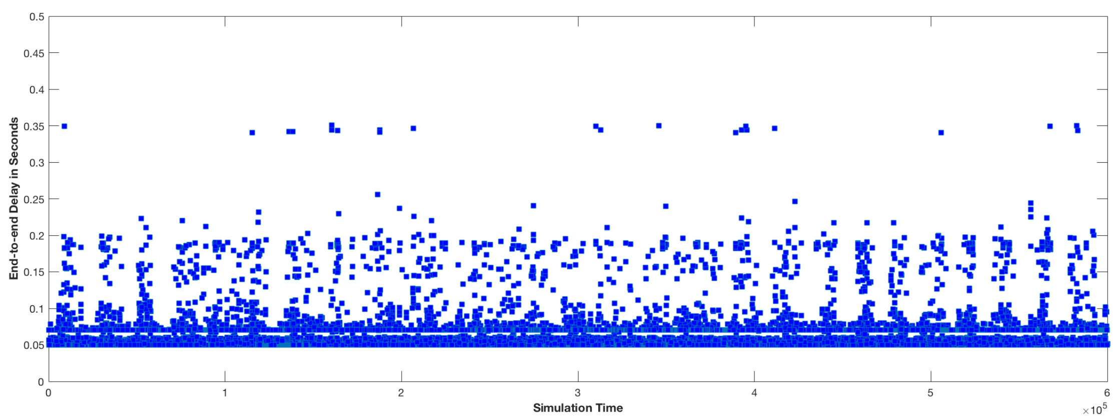


(b) Passed Up Packets from Origin Server to Server A vs. Server A to Origin Server, Number of IoT = 200x3

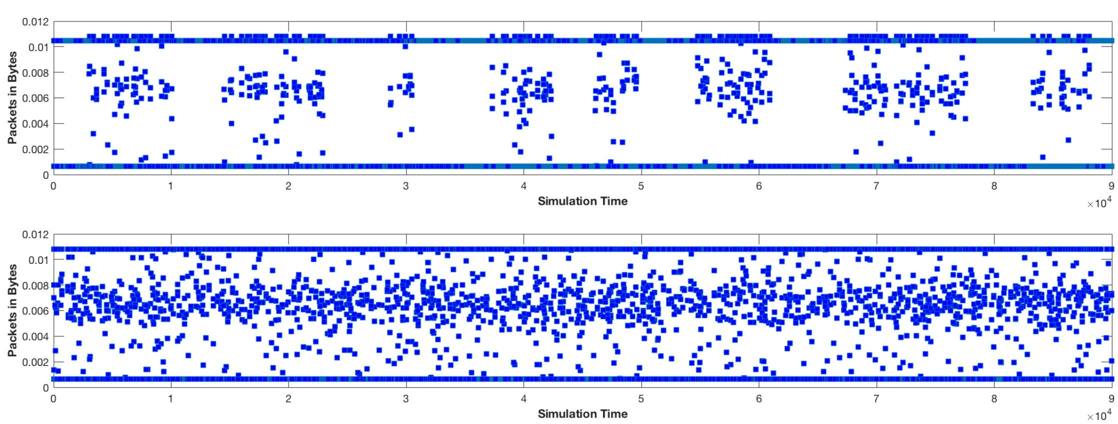
Fig. 6



(a) Bandwidth Measurements in Bits/sec For Number of IoT Devices 350x3



(b) End-to-end delay vs. simulation time. Number of IoT devices = 350x3



(c) Passed Up Packets from Origin Server to Server A vs. Server A to Origin Server, Number of IoT = 350x3

Fig. 7