2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)

GPN Game Users Performance Data Gathering and Analysis by a Custom-Built Tool

Nick McDonald, Corey Frank and Youry Khmelevsky*
Computer Science, Okanagan College, Kelowna, BC Canada
Emails: nick.mcdonald.94@gmail.com, corey.frank@myokanagan.bc.ca and ykhmelevsky@okanagan.bc.ca

Rob Bartlett and Alex Needham
WTFast, Kelowna, BC Canada
Emails: {rob, alex}@wtfast.com

*Also Affiliated with Mathematics, Statistics, Physics, and Computer Science Department
Irving K. Barber School of Arts and Sciences, UBC Okanagan, Kelowna, BC Canada

***Abstract*—This paper discuss the project results of the centralized storage for the client generated statistical data. The project aims was to fill the following requirements: (1) a new web API where local client statistics can be reported to a highly scalable and redundant central database, (2) to enable a user to easily save/share their connection results on a separate webpage, or with a widget that could be easily embedded on a forum page, and (3) to produce aggregate reports suitable for infographics, generated from the data reported by all users. Sample reports could be filtered by game, ISP, region, or some combination thereof.**

I. Introduction

Currently the WTFast's Gaming Private Network® (GPN®) [16] client gathers data about the internet connection, such as the latency, route and the amount of data sent/received. This can be displayed to the user at the end of a session allowing them to compare their connection using the GPN® to their regular internet connection but has no persistence. This means that users cannot easily look back on previous session's statistics or share their results with others. Furthermore, the data is not reported to WTFast so they have no way to perform meta-analysis on their user connection statistics. Without this data it can be difficult to the effectiveness of the GPN® system is or what kind of improvement users can expect when using the GPN®. A successful solution would reliably store a user's session data and display it back to them in a way that is understandable to the average user and be easily shareable to social media networks, online forums and webpages. It would also allow WTFast to perform analysis to create infographics to showcase the performance of their GPN®.

The goal of this project is to provide users a way to view statistical information about their network connections with and without using the WTFast's GPN®. It includes the collection and analysis of network data (pings, latency) and visualization of the analyzed data to provide meaningful information to the user, and allow the user to share his/her network statistics on forums and other social media. The WTFast's GPN® is a virtual private network that is configured specifically for gamers to provide an optimized online gaming experience. The data is collected by the WTFast's client software and sent to a back-end database through a web API. The data is analyzed and presented to the client via a web interface.

Users of this system will be the online gamers who are using, or testing the WTFast's GPN® and want to know how much performance gain they are getting by using it. Users will not have to be experts in computer networking, but should understand some of its key concepts such as latency, jittering, etc. to better understand the information presented to them. The user's interaction with the system will be limited, it will be largely automated, only with a few options select the session that they want to view and to generate an image that summarizes the data that can be embedded in forums or social media.

Our main contributions are: (1) a unique centralized storage solution for the client generated statistical data, which can be used to perform data analysis of network game applications, as well as to monitor the performance of game servers within a proprietary GPN®; (2) a new web API where local client statistics can be reported to a highly scalable and redundant central database.

Two networking and server optimization research projects GPN-Perf1: Investigating performance of game private networks" (2014) [1], [3], [2] and GPN-Perf2 research project application (2015-2018) were supported by Natural Sciences and Engineering Research Council of Canada (NSERC). This research potentially has far reaching impacts not only in reducing game latency but also in optimizing other types of network traffic via the prioritizing of the most important data packets sent over the network.

II. Background and Theory

As it was stated in [11], the "large-scale network simulation is an important technique for studying the dynamic behaviour of networks, network protocols, and emerging class of distributed applications, including Peer-to-Peer and Grid applications". We already discussed the GPN® infrastructure prototype in [3], which was developed to investigate traffic and game server behaviour in an artificially created testbed by using a popular game called Minecraft, by using a simple bot that was found online. GPN® is a WTFast's client/server solution to make online games faster by increasing game speed, reducing games disconnections, deviations and lag caused by spikes in packet traffic [16]. Our goal was to generate realistic network traffic related to an online video game environment by using game emulation applications available on the Internet.

In [15] we can find a definition of a *video game network*,

978-1-4673-8721-7/16/$31.00 ©2016 IEEE

which is “a distributed set of apparatuses which are capable of exhibiting an interactive single identity game”. Response times and network latencies are very important video game parameters, which can be a reasons for gamers frustration and dissatisfaction, especially in the multi-user environment. On the other hand, we can find in research papers, that “the on-line service’s computers themselves introduce latencies”, which are bigger with the bigger number of active gamers [14].

In [8] and in [9] discussed multiuser online video game architectures, specifically ways to reduce bandwidth and the servers processing time. Their approach may improve scaling, but it can lead to possible cheating, because gamers have access to distributed events and game states. A Hybrid Peer-to-peer (P2P) Architecture with a central arbiter was proposed in [13].

Authors in [18] discovered that latencies and delays are important in First-Person Shooter (FPS) games and it can give advantages to a gamer with less delays. “The time the information reaches the server that matters, not the time the player actually pushes the button”, stated in [3]. Other researchers have discovered that the latency in Internet is stronger for the First-person perspective online video games [4]. A study in [6] shows that the client’s traffic has “almost constant packet and data rate” in different First-person games.

Many authors tried to emulate and investigate game traffic for online games over the Internet. In [10] the authors investigate problems related to the StarCraft II game. They found network problems related to delays, jitters and packet losses. The traffic characteristics and traffic models are discussed, along with the suggested simulation and emulation tools. In [5] authors investigate traffic, generated by a thin client of OnLive games. They investigated traffic characteristics such as the bitrates, inter-packet times as well as packet sizes. In [12] the authors discussed Massively Multiplayer Online Gaming problems, especially related to data centre computing resources, and the related workload models. They tried to predict the resource demand by using trace-based simulation as well. In [17] authors discovered that gamers with more delays due to either a longer distance to servers, connection problems, or congestions have disadvantages to compare with the other gamers. The authors used bots instead of gamers to evaluate eliminating delay differences between players [17]. A traffic generation tool OpenAirInterface Traffic Generator (OTG), which can be used for online gaming testing and evaluation, as well as for modelling and simulation was discussed in [7].

III. System Design

We began our project with the search for an existing program that would meet our criteria. During the ongoing research project we have designed a system made up of several parts. The system’s Domain Model is shown at Fig. 1. Architecture consists of a web API built on C# .NET to allow the data generator or client to post data to our back-end Elasticsearch database. As well as a website for viewing and sharing their analytics. The Architecture of the developed system is shown at Fig. 2.

The web API Class Diagram is shown at Fig. 3. A method for posting data to the Elasticsearch database, and another that allows the website to retrieve data from it.

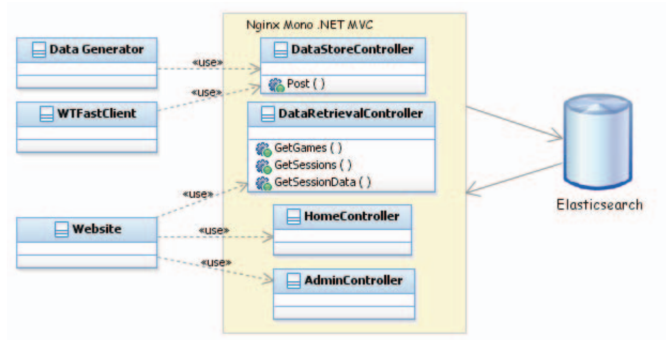

Fig. 1. System Domain Model

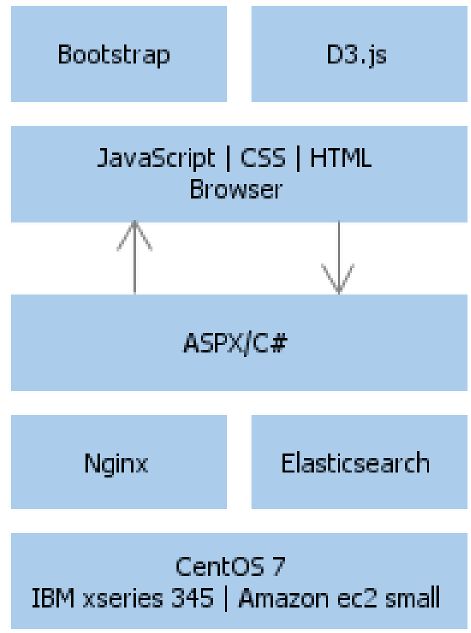

Fig. 2. System Architecture

IV. Development

Additionally a data generator was developed. This is a small Java application used to generate data to test the web API and to populate our database. It generates all of the fields that the WTFast’s client would supply in the real system, but is configurable so we can test with large or small volumes of data from many virtual users.

Fig. 4 shows the logical structure of the data. Each user creates a new session each time they play a game, and each session consists of many data packets which contain the network statistics using WTFast, and without collected every few seconds as the user plays.

A. *User’s Interface*

The prototype of the client side application is shown at Fig. 5. It is easy to use. The user logs in with their WTFast account and is shown their most recent session. Two dropdown menus let the user select session by game and date and time of the session. The graph shows the ping latency for the whole

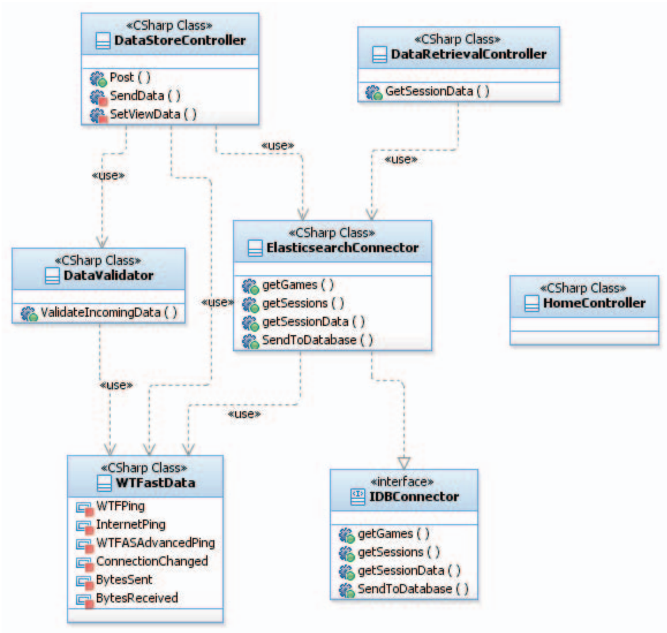

Fig. 3. Web API Class Diagram

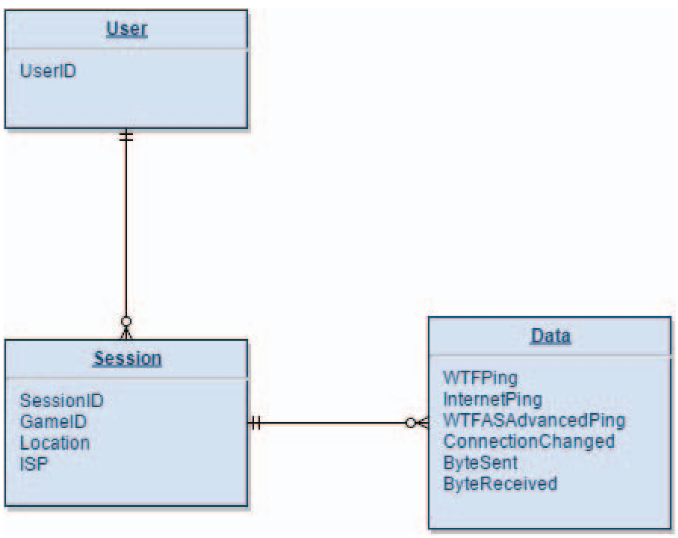

Fig. 4. Entity-Relationship Diagram

session, red shows the Internet ping, and yellow shows the ping using the WTFast's GPN® which is lower and contains less spikes. Aggregate data is shown below the graph with green showing the connection improvement using the GPN®. The save button lets the user save the data as an image that summarizes the data and provides a link that can be used to embed the image in a forum.

V. Future Work

Our future research will be related to utilizing the infrastructure created in the first phase of the project as framework on which to test a live version of the system to be integrated with WTFast's current websystem. This infrastructure will provide the tools necessary to analyze game data and provide a way of viewing and analyzing the game servers on the GPN®.

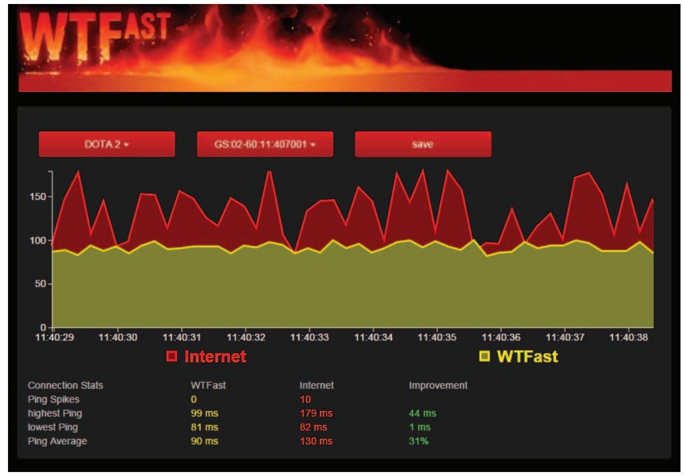

Fig. 5. User Page Prototype

This data will be used to help further increase the performance of the GPN®.

Using our system, we will be able to gather information that could be used to further enhance the GPN® that would have gone otherwise unnoticed and unused. This information will be viewed by infographics generated by Kibana, using many different metrics and algorithms will allow the ability to predict future events that could affect the GNP; and be able to use this information to prepare for otherwise unavoidable events.

Testing of the API will first be done by using several instances of the generator to test the system under heavy loads. The purpose of these tests will be to see which part of the system will cause an issue when put under a heavy load. This will let us know which part of the system will need improvement before going live. Once the system performs efficiently under testing conditions it will be implemented in the WTFast beta client for trial integration. During this time, we will observe how the system behaves under real conditions; and implement any necessary changes to fix any issues. Once the system is fully integrated and working properly in the WTFast's client we will then be able to start observing real data and begin working on ways to take the information we receive from the users and display it using infographics in a manner that will be useful for the client.

Future research will be aimed at taking the data from the users and using that to create infographics that can be used to assess the key vulnerabilities in the GPN®. This information will then be evaluated and then changes could be made to the GPN® to increase its performance by predicting things like future traffic spikes or to examine possible external factors that are affecting the GPN®.

VI. Conclusion

Network analysis is a broad spectrum of research that can be studied in many different ways. We have set up a system that allows monitoring of client data from a robust data centre. The API was developed to accept data being posted to it from a client, but while in construction a generator was built to simulate data to be sent. This gave us the option to control the data going into the database as well as simulate large amounts of data being sent at a time. The data is reported to the Elasticsearch database and can be viewed by the user from the website interface that contains graphs and information about their game session. The project was completed to its specifications, in that all the functionality of the system has been built. We anticipate that his project will be a useful tool to use when gathering and analyzing client network statistics. Based on our experiments, our infrastructure gives to us a suitable tool for the network performance investigation and to generate a stress test for the system.

ACKNOWLEDGMENTS

The described WTFast's capstone project was developed by Computer Science students at Okanagan College within COSC 470 Software Engineering and COSC 471 Software Engineering Project courses at Bachelor of Information Systems program, Computer Science department in Fall 2015 and Winter 2016. We would like to thank to COSC 470/471 students, who participated in the project development: Jon Ohlhauser, David Leader and Cheng-Kao Chiang.

We would like to thank NSERC's CCI Engage College program for supporting ARD Level 1 "GPN-Perf1: Investigating performance of game private networks" in 2014 and ARD Level 2 "GPN-Perf2" research project, which is just started in February 2016. Our thank to Amazon Web Services, Inc. for supporting our capstone student software engineering and our student research projects by AWS Grants for Research and Education.

Our thank to Amazon Web Services, Inc. for supporting our capstone student software engineering and our student research projects by AWS Grants for Research and Education, as well as to Atlassian for supporting us by community licenses for Jira project management and Bitbucket cloud tools. Our thanks to IBM Education for supporting us by IBM Rational SW Architect licenses in our system design and SW engineering projects.

REFERENCES

[1] T. Alstad, J.R. Duncan, S. Detlor, B. French, H. Caswell, Z. Ouimet, Y. Khmelevsky, G. Hains, R. Bartlett, and A. Needham. Minecraft computer game performance analysis and network traffic emulation by a custom bot. In *Science and Information Conference (SAI), 2015*, pages 227–236, July 2015.

[2] T. Alstad, J. Riley Dunkin, S. Detlor, B. French, H. Caswell, Z. Ouimet, Y. Khmelevsky, and G. Hains. Game network traffic simulation by a custom bot. In *Systems Conference (SysCon), 2015 9th Annual IEEE International*, pages 675–680, April 2015.

[3] Trevor Alstad, J. Riley Dunkin, Rob Bartlett, Alex Needham, Gaétan Hains, and Youry Khmelevsky. Minecraft computer game simulation and network performance analysis. In *Second International Conferences on Computer Graphics, Visualization, Computer Vision, and Game Technology (VisioGame 2014)*, Bandung, Indonesia, November 2014. Accepted for publication.

[4] Mark Claypool and Kajal Claypool. Latency and player actions in online games. *Commun. ACM*, 49(11):40–45, November 2006.

[5] Mark Claypool, David Finkel, Alexander Grant, and Michael Solano. Thin to win?: Network performance analysis of the onlive thin client game system. In *Proceedings of the 11th Annual Workshop on Network and Systems Support for Games*, NetGames '12, pages 1:1–1:6, Piscataway, NJ, USA, 2012. IEEE Press.

[6] Johannes Färber. Traffic modelling for fast action network games. *Multimedia Tools and Applications*, 23(1):31–46, 2004.

[7] Aymen Hafsaoui, Navid Nikaein, and Christian Bonnet. Analysis and experimentation with a realistic traffic generation tool for emerging application scenarios. In *Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques*, SimuTools '13, pages 268–273, ICST, Brussels, Belgium, Belgium, 2013. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[8] Takuji Iimura, Hiroaki Hazeyama, and Youki Kadobayashi. Zoned federation of game servers: A peer-to-peer approach to scalable multi-player online games. In *Proceedings of 3rd ACM SIGCOMM Workshop on Network and System Support for Games*, NetGames '04, pages 116–120, New York, NY, USA, 2004. ACM.

[9] Jared Jardine and Daniel Zappala. A hybrid architecture for massively multiplayer online games. In *Proceedings of the 7th ACM SIGCOMM Workshop on Network and System Support for Games*, NetGames '08, pages 60–65, New York, NY, USA, 2008. ACM.

[10] Choong-Soo Lee. The revolution of starcraft network traffic. In *Proceedings of the 11th Annual Workshop on Network and Systems Support for Games*, NetGames '12, pages 18:1–18:2, Piscataway, NJ, USA, 2012. IEEE Press.

[11] Xin Liu and Andrew A. Chien. Traffic-based load balance for scalable network emulation. In *Proceedings of the 2003 ACM/IEEE Conference on Supercomputing*, SC '03, pages 40–, New York, NY, USA, 2003. ACM.

[12] Vlad Nae, Alexandru Iosup, Stefan Podlipnig, Radu Prodan, Dick Epema, and Thomas Fahringer. Efficient management of data center resources for massively multiplayer online games. In *Proceedings of the 2008 ACM/IEEE Conference on Supercomputing*, SC '08, pages 10:1–10:12, Piscataway, NJ, USA, 2008. IEEE Press.

[13] Joseph D. Pellegrino and Constantinos Dovrolis. Bandwidth requirement and state consistency in three multiplayer game architectures. In *Proceedings of the 2Nd Workshop on Network and System Support for Games*, NetGames '03, pages 52–59, New York, NY, USA, 2003. ACM.

[14] S. G. Perlman. Network architecture to support multiple site real-time video games. United States Patent number 5,586,257, Dec. 17, 1996.

[15] D. H. Sitrick. Video game network. United States Patent number 4,572,509, Feb. 25, 1986.

[16] AAA Internet Publishing Inc. WTFast. http://www.wtfast.com.

[17] Sebastian Zander, Ian Leeder, and Grenville Armitage. Achieving fairness in multiplayer network games through automated latency balancing. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, ACE '05, pages 117–124, New York, NY, USA, 2005. ACM.

[18] Qili Zhou, C.J. Miller, and Victor Bassilious. First person shooter multiplayer game traffic analysis. In *Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on*, pages 195–200, May 2008.