A New Online Tool for Gamer Network Performance Analysis

Nick McDonald, David Leader, Cheng-Kao Chiang and Youry Khmelevsky
Computer Science, Okanagan College, Kelowna, BC Canada
Emails: nick.mcdonald.94@gmail.com, solorak@gmail.com and fan119053@gmail.com, ykhmelevsky@okanagan.bc.ca

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

Also Affiliated with Computer Science, UBC (Okanagan Campus), Kelowna, BC Canada

***Abstract*—This paper discusses a new online tool for online gamers to share networking performance results and to produce aggregate networking performance reports generated from the data reported by all users. It is necessary to improve the quality of service and make online games faster. This will help improve latency sensitive game data collection. It will help to improve gaming performance through the global network of servers, optimizing the game connection from end to end.**

A Web API was developed to receive data in the form of a HTTP post from WTFast clients while they are playing their games. A data generator was also built to simulate thousands of concurrent users posting data to the web API. The web API stores the data in a Elasticsearch database. A web interface was built for WTFast clients to view their historical network data and to share it on forums or social media. The web interface shows a graph of their network performance with and without using the GPN^R. The complete system is anticipated to need to handle approximately 400,000 concurrent users sending network data every 2 seconds.

I. Introduction

The current "WTFast's Gaming Private Network^R (GPN^R)" [1] online tool "gathers data about the internet connection, such as the latency, route and the amount of data sent/received" [2], but gamers still can't see the comparison of the regular internet connection vs of the improved connection by the GPN^R. Moreover, it would be nice for gamers and for the WTFast system administrators to see the historical information of the connection to improve the provided services.

Users should be able "to view statistical information about their network connections with and without using the WTFast's GPN^R. 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^R 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" [2].

Our research contributions are:

a) A new online tool for the online gamers, clients of the WTFast to share networking performance results and to produce aggregate networking performance reports generated from the data reported by the gamers.
b) A new data generator for the online tool and for the Elasticsearch database testing.
c) A proof of a concept for the networking metadata collection of the online gamers.

II. Related Works

The "large-scale network simulation is an important technique for studying the dynamic behaviour of networks, network protocols, and an emerging class of distributed applications, including Peer-to-Peer and Grid applications" [3]. The original GPN^R infrastructure prototype, which was built in 2014 for initial tests we discussed in [4]. GPN^R makes online games faster by increasing game speed, reducing game disconnections, deviations and lag caused by spikes in packet traffic [2].

A "*video game network*, 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" [5].

"The online service's computers themselves introduce la-

or congestions have disadvantages to compare with the other gamers. The authors used bots instead of gamers to evaluate eliminating delay differences between players" [12].

"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 described in [13].

III. System Design

At the start of our project we gathered requirements for our system, decided what technologies we were going to use, and what features our software would have. The architecture we decided on consists of a data generator built on Java, web API built on C# .NET MVC, and Elasticsearch as our backend database. The web API allows the data generator (or WTFast client) to post data to our back-end Elasticsearch database and also hosts a website for viewing and sharing summaries of their data.

Fig. 1 shows a layered view of the system. Users interact with the system via web browser for viewing their data. The browser interprets the JavaScript, HTML and CSS that comes from our C# .NET MVC middleware. We use a web framework named Bootstrap to display our webpage in a nice, responsive way, and the D3.js library for graphing the network data. Our middleware is written using C# .NET MVC, which runs on our Nginx web server. Because .NET applications are ment to run on Windows, to run them on CentOS 7 we had to run the application on Mono, which is a framework that allows .NET applications to run on Linux. We used FastCGI to host the application on top of Nginx. We used Elasticsearch as our database. Elasticsearch is a NoSQL database, we chose to use

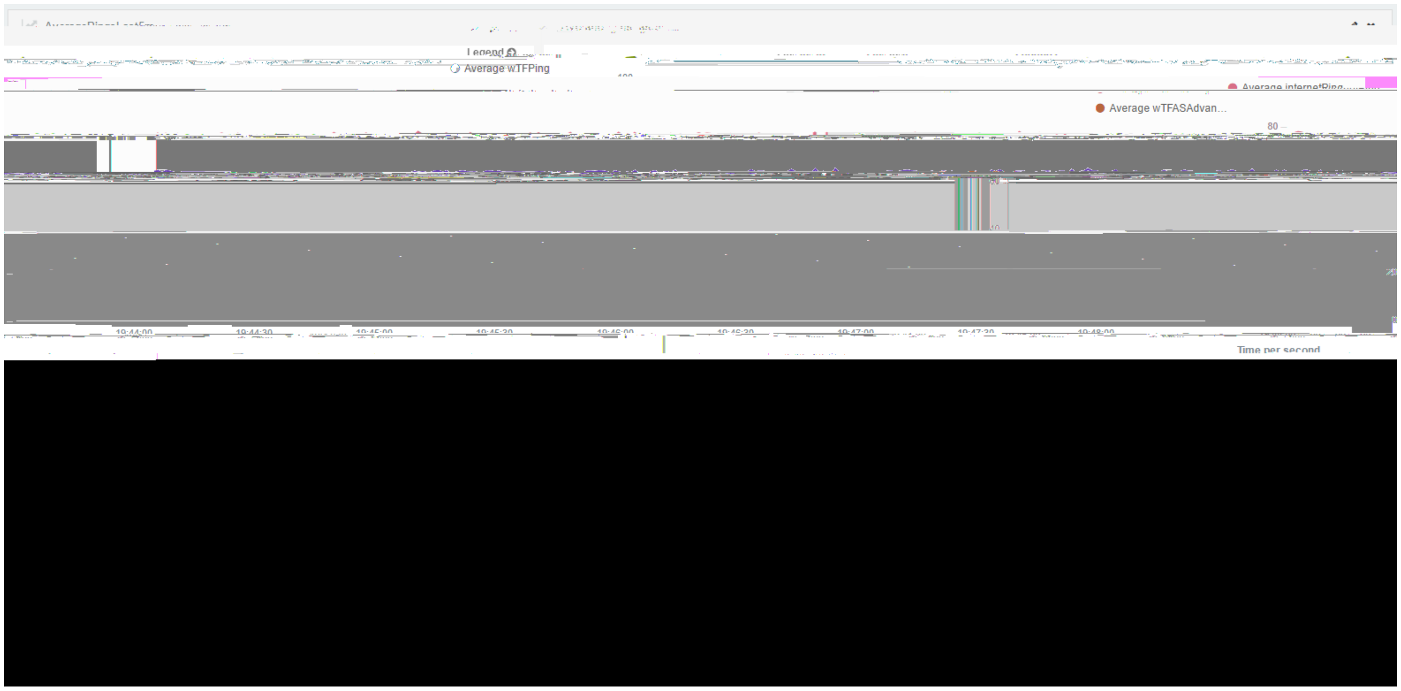

Fig. 2. The average of all pings sent to the database within the last five minutes

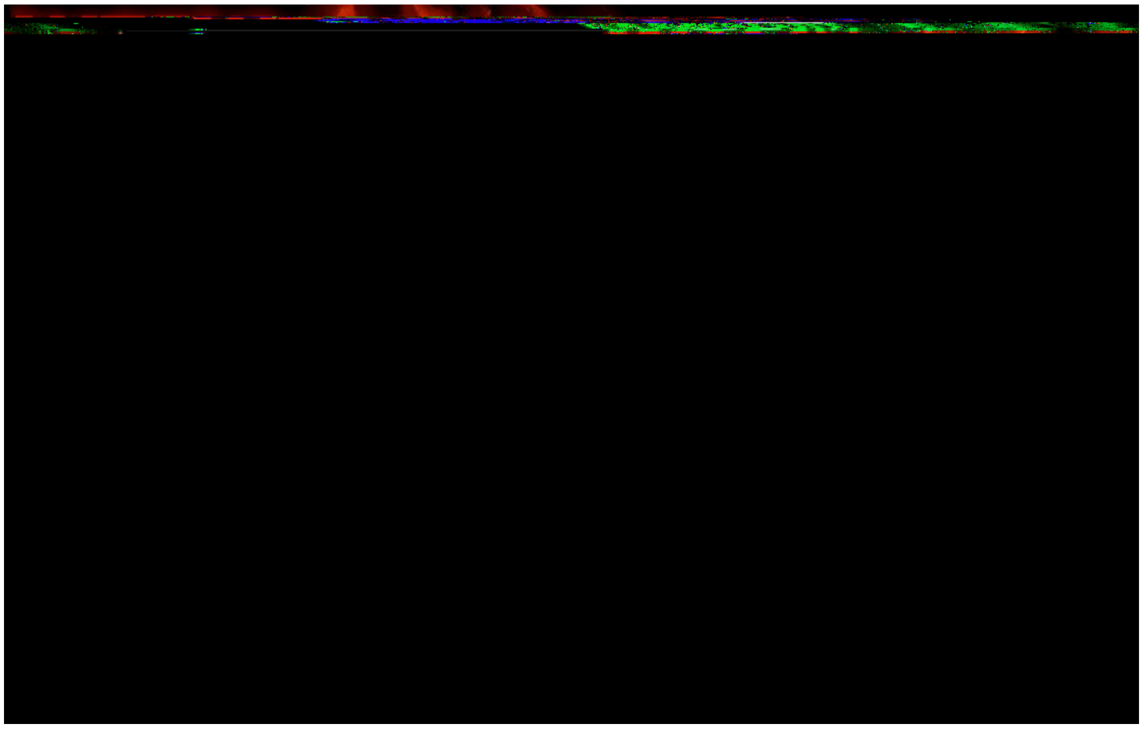

Fig. 3. Web Client Interface

ping spikes as well as a percentage improvement when using the GPN^{R}.

The last feature of the web interface is the "Upload your Speed" button. Clicking this button will generate an image similar to what is shown in Fig. 5 and will upload the image to the server. In a pop up window users will be shown the image, and be given the URL of the image so that they can embed it in forums or share it on social media.

Using the Bootstrap framework allowed us to make a responsive design that would work on many different screen resolutions including mobile phones as seen in Fig. 4.

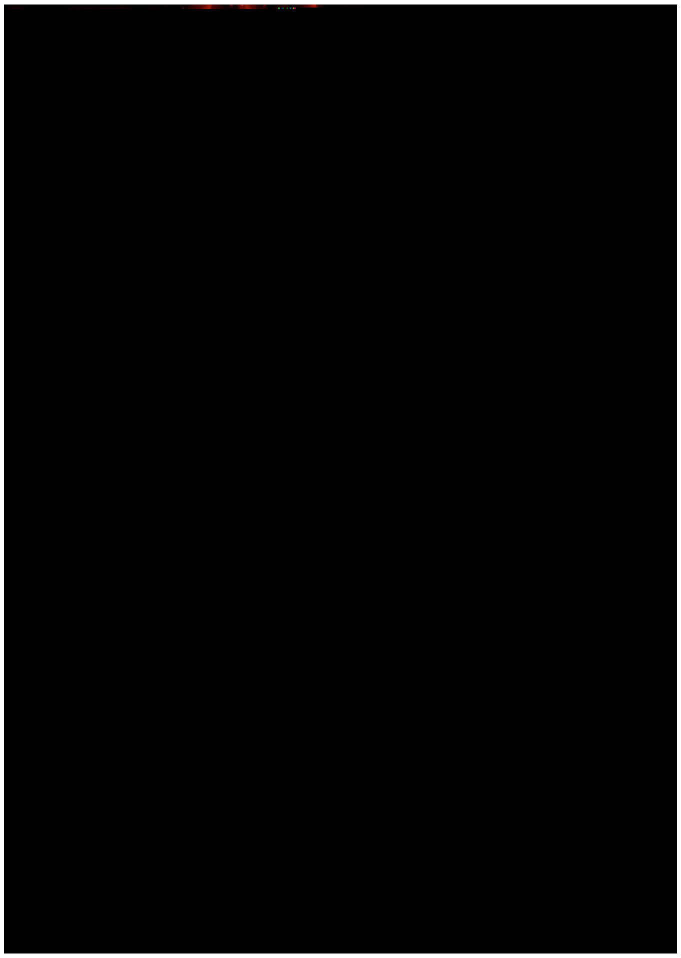

Fig. 4. Web Client Interface (Mobile Version)

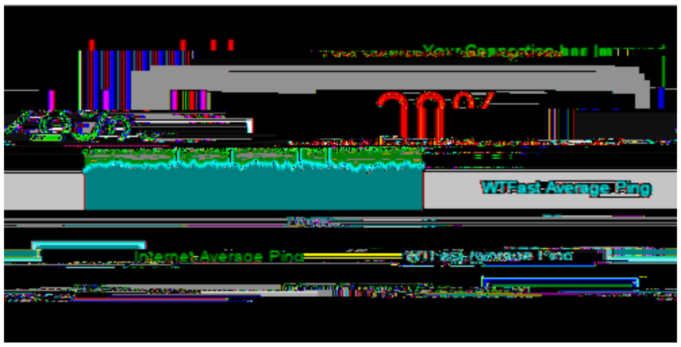

Fig. 6. Generator Class Diagram

We would like to thank to Amazon Web Services, Inc, Atlassian, IBM and all other companies, who supported our educational and research projects.

References

[1] A. I. P. I. WTFast., "http://www.wtfast.com."

[2] N. McDonald, C. Frank, Y. Khmelevsky, R. Bartlett, and A. Needham, "GPN game user performance data gathering and analysis by a custom-built tool," in *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), 2016*, Vancouver, BC, Canada, May 15–18 2016, pp. 1099–1103.

[3] X. Liu and A. A. Chien, "Traffic-based load balance for scalable network emulation," in *Proceedings of the 2003 ACM/IEEE Conference on Supercomputing*, ser. SC '03. New York, NY, USA: ACM, 2003, pp. 40–. [Online]. Available: http://doi.acm.org.ezproxy.okanagan.bc.ca/10.1145/1048935.1050190

[4] T. Alstad, J. R. Dunkin, R. Bartlett, A. Needham, G. Hains, and Y. 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.

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

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

[7] Q. Zhou, C. Miller, and V. Bassilious, "First person shooter multiplayer game traffic analysis," in *Object Oriented Real-Time Distributed Com-*