



HAL
open science

The WTFast's Gamers Private Network (GPN ®) Performance Evaluation Results

Gaetan Hains, Chris Mazur, Jesse Ayers, Jack Humphrey, Youry Khmelevsky,
Ty Sutherland, Gaétan Hains

► **To cite this version:**

Gaetan Hains, Chris Mazur, Jesse Ayers, Jack Humphrey, Youry Khmelevsky, et al.. The WTFast's Gamers Private Network (GPN ®) Performance Evaluation Results. 2020 IEEE International Systems Conference (SysCon), Aug 2020, Montreal, France. pp.1-6, 10.1109/SysCon47679.2020.9275855 . hal-04031723

HAL Id: hal-04031723

<https://hal.science/hal-04031723v1>

Submitted on 14 Dec 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The WTFast's Gamers Private Network (GPN[®]) Performance Evaluation Results

Gaétan Hains

*Laboratoire d'Algorithmique, Complexité et Logique
Université Paris-Est Créteil
France, Paris
Email: gaetan.hains@gmail.com*

Chris Mazur

*Computer Science Department
Okanagan College
Kelowna, Canada
Email: christopheramazur@gmail.com*

Jesse Ayers

*Computer Science Department
Okanagan College
Kelowna, Canada
Email: jdeanayers@gmail.com*

Jack Humphrey

*Computer Science Department
Okanagan College
Kelowna, Canada
Email: jackrbhumphrey@gmail.com*

Youry Khmelevsky

*Computer Science Department
Okanagan College
Kelowna, Canada
Email: ykhmelevsky@okanagan.bc.ca*

Ty Sutherland

*COO
WTFast
Kelowna, Canada
Email: ty@wtfast.com*

Abstract—The WTFast Gamers Private Network (GPN[®]) is a client/server solution designed to improve network conditions with online games that utilize centralized game servers. Using a GPN[®], connections between the clients personal computer and game network are improved in terms of reduced network latency, spikes, and jitter.

A common concern among players participating in online games is the game's connection stability. A key aspect of WTFast's GPN[®] service is its greatly improved connection consistency. This is accomplished by routing data from the client to the game server in a more controlled manner.

To determine what kind of conditions and improvements can be found using a GPN[®], we gathered data from thousands of GPN[®] and non-GPN[®] (regular internet) connections to create a statistical model of spike predictability, frequency, and severity using Markov Chains. Paired with a comparison of latency, spikes, and jitter across GPN[®] / non-GPN[®] lines, we provide a statistically sound report on the efficacy of WTFast's GPN[®] service.

I. INTRODUCTION

The key objective of our work is to understand the evolution of network latency and its features relating to that network's quality of service, and to determine the efficacy of GPNs[®] compared to regular internet. To do this, we looked at thousands of data points from connection records made between source and destination servers in both GPN[®] and non-GPN[®] environments. We analyzed the differences in GPN[®] and non-GPN[®] connections using graphs, tables, and Markov Chain machine learning models.

An initial analysis of emulated GPN[®] connections published in 2017 [1] showed promising evidence for the superiority of GPNs[®] with regard to network consistency. The results from those experiments in controlled, simulated environments are what led us to gather and analyze a larger data set of network connections. In total, we gathered 100,000 records of GPN[®] connection data, and a further 30,000 records of non-GPN[®] connection data to be analyzed and compared.

A major consideration for our analysis is the context and business use of GPNs[®]. Their use in online videogames to reduce frustration and increase competitiveness is the main attraction to users. Consistently low latency, low spike and jitter frequency, and low spike and jitter severity are what users expect from a GPN[®] service.

II. EXISTING WORKS

Unsurprising and sub-second reaction time is a key worry for intuitive PC frameworks [2]. For most video games this is an undeniable prerequisite that cutting-edge equipment has fulfilled, regardless of the increase in those games' demands for more complex interactions and improved graphical quality.

A *video game network* is a distributed set of “apparatus which [is] capable of exhibiting an interactive single identity game,” as defined in a patent dated 1986 [3]. The requirements for response time are even more stringent in this context and in addition to inevitable network latencies, “the on-line service's computers themselves introduce latencies, typically increasing as the number of active users increases” [4]. The work completed by the previous iteration of the project [1] consisted of a test examination of the conditions for fulfilling this key prerequisite, particularly in low and unsurprising reaction time for a game system aimed toward a varying quantity of players.

The past fifteen years have seen a developing enthusiasm for handling this issue. A few analysts like Iimura [5], Jardine [6] and co-creators have suggested shared structures for multiplayer online video games with the expectation of decreasing the data transfer capacity and preparing perquisites on servers. This has the potential of better scaling, but “opens the game to additional cheating, since players are responsible for distributing events and storing state”. Pellegrino et al. [7] have then proposed a hybrid architecture called P2P with central arbiter. The transmission capacity necessities on the

router are lower than the server of a unified design. In the same way as other non-utilitarian properties of online systems (security, adaptability, unwavering quality and so on) the decision among centralization and appropriation isn't one that can be offered a complete response. Ward et al. [1] focused on a legitimately unified architecture that has capability for consistency and versatility of the server and router execution.

Other studies [8] have contemplated similar execution issues within the sight of versatile player hubs. Regardless of its significance for the future, this line of focus shows it is even less developed than the P2P approach.

Performance problems studied by [8] in the presence of mobile player nodes reflect this. Despite its clear importance for the future, this mobile architectures appear even less mature than the P2P approach.

Zhou, Miller, and Bassilious [9] have made the obvious but central observation that "Internet delay is important for FPS games because it can determine who wins or loses a game." Many game mechanics are incredibly time sensitive, yet it is the time data is processed by the server that matters, not the time the player actually pushes the button. Our tests measure inter-packet times as they have in their analytical model. Those creators' examination additionally considered the impacts of other internet traffic. Our examination presupposes the impact of other internet traffic as statistically distributed across the data, and uses a large amount of data to dilute its effect.

Claypool and Claypool [10] have seen that Internet latency's impact is most grounded for games with a first-person point of view and an evolving model. The project work [1] portrayed trying different tasks with the game Minecraft which is first-person and has changing game conditions.

Later investigations [11], [12] of first-person shooter games have demonstrated time arrangement conduct of game traffic and tried the model on up to eight distinct games. Such a near report would not have enabled the project to get truly steady load estimations, thus the decision of a solitary first-person game [1]. The investigation of Wu, Huang and Zhang [13] demonstrates that "the server-generated traffic has a tight relationship with specific game design," again affirming the requirement for exact specifications of a given design on a solitary game. Hariri et al. [14] go into more detail by planning a model of the player's action to control for traffic designs.

The implementation of a Latency Management System (LMS) is another solution that gets implemented into game networks. Inconsistent latency poses a concern for developers who wish to create a fair game environment where all players can share in a similar experience. However, due to poor network traffic conditions, how players interact with the game and one another can be significantly hindered by the traffic conditions of others. To compensate for this issue, there are many different mechanisms that exist to conceal the problems created by varied network conditions which can be implemented depending on the needs of the developer [15]. These mechanisms take packet loss, jitter, and server delay into account to attempt to minimize the adverse effects that can occur. For example, shot behind cover lag which can

be encountered when a player with sufficiently poor network condition interacts with where another player was rather than where they currently are in relation to their own game client. In such a case, the game acknowledges the client-side interactions of the lagging player [16] which leads to poor experiences for players.

One LMS solution is to group players with similar network conditions together [17]. This allows players with stable network conditions to receive an optimal experience as the affects of latency in highly competitive environments can heavily affect performance [18]. Another solution is the prospect of optimizing data flow at the network level [19]. This solution can provide positive results for slower networks but risks stressing the unfairness between flows if network congestion is severe. These types of solutions go beyond what was hoped to address in their paper [1], but understanding the need for this information is important for the growth of gaming network solutions.

"A study of different first-person games shows that the client traffic is characterized by an almost constant packet and data rate" [20]. The study found that "the average inter-packet time for client to server traffic to be 51ms for the game being studied". The bot system created in [1] sends its action packets at 50ms intervals [21] in order to better observe how the networking design affects latency.

As it was shown in [22], the "bottleneck in the server is both game-related as well as network-related processing (about 50%-50%)". In [1], the examination done generally focused on the servers' exhibited improvement, the system traffic investigation [23], and the execution of a custom bot for Minecraft [21]. During this exploration, the most noteworthy remaining task at hand for the CentOS 6.5 virtual server was examined by using a custom-created bot for Minecraft.

A few creators talk about intelligent web based games, particularly games that are identified as a "first person shooter (FPS)" [11], [12] and discuss system traffic associated with such games [9] such as system effects on the games and reasonable traffic generators. The goal of [1] was to be able to emulate the server traffic of up to a thousand or more players. Latency is another test for online games as it is reported in [10], [7], and [24] and is a significant factor of an online gaming experience. Ward et al. [1] constructed their framework to simulate the latency delays of a rush hour gridlock [23].

In a paper written by Jardine et al., [6] "massively multi-player online games with a client-server architectures and peer-to-peer game architectures" are investigated. The creators of these architectures built a hybrid game architecture to diminish game server data transfer capacity. In the paper written by Iimura et al. [5], their findings included that creators even proposed to execute a zoned organization model for the multi-player online games attempting to lessen load on the game servers. A US 5956485 patent [4] portrays how to interface various remote players of online games on a conference phone line in a way which could lessen latency for the game players. This concept has been used and expanded upon with more modern internet technologies.

In the technical report from IBM [2], it was demonstrated that “rapid system response time, ultimately reaching sub-second values and implemented with adequate system support, offers the promise of substantial improvements in user productivity” and it is even better to “implement sub-second system response for their own online systems.” They also referenced that not all online PC frameworks are adjusted. Additionally, they isolated framework reaction time for two huge gatherings: PC reaction time and correspondence time which are both required for the game players client experience.

Within the medium of online gaming, there is a strong continuing trend of Free-to-Play models, of which many different online games have millions of subscribers and hundreds of thousands of concurrent players [25], [26], [27]. Due to the nature of how many of these games fragment their players for individual game sessions, a popular model discussed is the concept of a hybrid peer-to-peer (P2P) network architecture. This architecture ensures that players within reasonable proximity and network characteristics are paired together by a latency management system [17], [28] but also has the connection of the players rely on a centralized server to ensure game integrity. Within a fully P2P game environment, the absence of a centralized server results in all crucial data comes from the game host which has the potential risk of the host deliberately cheating or sending malicious data [29] which can have varying degrees of consequence for all connected players.

Hybrid P2P servers rely on a large web of edge servers [30] that take advantage of their large regional diversity to help minimize latency between the centralized servers and players, and reduce excess strain.

Cloud gaming, or gaming on demand, has become another popular alternative for creating network architectures for online games. MMOs are regularly turning to cloud gaming as a network solution as the number of concurrent players climbs into the hundreds of thousands [25]. Cloud gaming offers a scalable solution that handles large changes in players while helping manage the cost of bandwidth consumption [31]. This has become an increasingly popular medium for online games but comes with a certain level of stigma as the medium has been surveyed many times and customers are wary of the drawbacks [32]. The biggest risk involved when choosing to implement cloud gaming is that the Quality of Experience (QoE) comes with a large set of challenges stemming largely from challenges regarding latency [33], [34]. Competitive online games that rely on minimizing latency and packet time often avoid cloud networks as even current Inter-player Delay Optimization (IDO) solutions only serve to reduce the perception of response delay from players rather than eliminating it [35], [33]. However, cloud gaming strongly serves certain Massively Multiplayer Online (MMO) Games; while the experiences of players in First Person Perspective games suffer more from perceived latency issues, many other genres do not.

III. MODEL ELEMENTS

In this section, the mathematical model elements are detailed. For modeling the time series, Markov chains were used and followed the work of Mallick-Hains-Deme [36] for predictive monitoring.

Each time-series vertical axis is divided into a sequence of intervals from its lowest possible value to its largest possible value. We consider the observed time-series as a trace of a stochastic change of vertical level such as the evolution of a Markov chain. Following the classic textbook [37] section 10.6, we estimate the transition probabilities of this Markov chain by the observed frequencies of state changes.

For example, if the value is a percentage we could divide its vertical axis into 10 buckets $[0\%, 10\%]$, $[10\%, 20\%]$, \dots , $[90\%, 100\%]$. With this setup, we can estimate the Markov Chains time-series transition matrix, allowing us to produce more random, but realistically similar, time-series. In our example, the Markov Chains transition matrix M is 10×10 and $M(i,j)$ and the transition probability from state i to state j is estimated by the fraction of observed transitions $M(i, _)$ that have j as a destination. Such things that can be determined by tracing one-dimensional random walks with the matrix are:

- 1) Computing the steady-state probability of being in each state and estimating the distribution of states by the frequency that they are found in the time-series
- 2) Drawing an initial state randomly with probability distribution given by 1. Call this state $X(t) = i$.
- 3) Drawing the next state $X(t + 1)$ with probability distribution given by $M(i, 1), M(i, 2), \dots, M(i, n)$.

Many other things can be computed from the matrix, but for the purpose of our initial model this will be enough. From an observed time-series we can compute its estimated Markov chain matrix M . From M , an infinite variety of similar time-series can be cheaply and easily computed.

IV. MEASUREMENT DATA

Our previous research[1] analyzed and modeled game servers and players’ time behaviour. Among the conclusions, as confirmed by other research papers, we found that bandwidth had a limited effect on gaming experience but that latency was crucial. Our current research tasks therefore analyze the statistical and time-series behaviour of both GPN[®] and non-GPN[®] for online games. We summarize here our initial conclusions, their relevance for massive simulations, and possible improvements.

All data and analysis steps are replicated separately for GPN[®] and non-GPN[®] networks. In this manner we can measure the effect of using GPN[®] and also measure each type of network in detail. Our data sets are tables of traceroute measurements. Each row measures the delta of latencies involved in sending a signal to a server and getting a signal in return. Each measurement contains the source and destination IP addresses, a list of 1 to 25 hop deltas in milliseconds, whose sum is the total latency the or round-trip time of the traceroute. Table I below provides a representation of these data sets.

TABLE I
METADATA

	Records	Sources	Destinations	Hops
GPN [®]	29584	214	108	0 - 25
Non-GPN [®]	234	1	66	0 - 25

We analyzed 29584 such traceroutes for GPN[®] and 234 non-GPN[®]. As we will see, the small size of the non-GPN[®] data set did not appear to affect the stability of the results yet we are currently collecting massive data sets of this kind for further confirmation. The geo-locations of all IP addresses are available but have not been analyzed as a possible parameter to influence latency. Our previous work has related them (through distance) to delays in a clear manner [38] but the analysis we are now conducting is concerned with extreme delays (spikes) likely caused by traffic or servers, and not pure distance alone.

For GPN[®] there were 214 different source IPs and 108 destination coordinates so on average only about 1.2 traceroutes per source-destination pair. The average number of hops was 12. For non-GPN[®], there was only one source IP (our lab’s address) and 66 destination coordinates which amounts to about 3.5 traceroutes for every destination IP.

We have defined a spike as a hop delta that is “several” times the average latency delta in its traceroute; currently we define “several” as ten times the average delta. According to this definition, less than 0.1% of the GPN[®] traceroutes contain spikes, which is a clear confirmation of the advantage of this technology. By comparison, 40% of the non-GPN[®] routes contain spikes. While spikes occurring in GPN[®] routes is very rare, the severity of them is very high, with total latencies ranging between 61 and 164 milliseconds. By comparison, total latencies of non-GPN[®] routes with spikes varies from 19 to 61 milliseconds. This phenomenon will require further analysis, and avenues for investigation based on time-of-day, network traffic, IP addresses, router configuration, or other parameters of the network are available.

We then used the Markov model outlined above to analyze the time-series behaviour of the traceroutes; we aim to categorize them regardless of whether this was from their internal sequence of delays or external factors. We thus partitioned delays into four “states” which consisted of very low, low, high, or spike. From each data set’s global statistics we defined a Markov matrix of transition probabilities; how likely it is that the next delay will be of type j , given that the previous one was of type i . The Markov matrix for GPN[®] data reaches a stable state at matrix power 8 as seen in Table II.

The same stable state for non-GPN[®] routes is reached after only 4 hops (matrix power 4) as seen in Table III.

TABLE II
GPN[®] STABLE STATE MARKOV MATRIX

Low	Medium	High	Spike
49.7%	48.7%	0.5%	0.9%

TABLE III
NON-GPN[®] STABLE STATE MARKOV MATRIX

Low	Medium	High	Spike
65.5%	12.4%	13.8%	8.2%

Tables II and III show that overall only 0.9% of hops will be spikes in GPN[®] tracerts, while hops on non-GPN[®] tracerts are nearly 10 times as likely at any given point to spike.

To compare those statistical distribution vectors (the fixed-points of Markov matrices) with their parent data set, we use them to predict which ones of the routes should contain a spike and then classified the predictions as true or false and positive or negative. We used 20% of data to create the matrices then tested that prediction on the remaining 80%. It turns out that the distribution of states is predicted by up to 0.2% for GPN[®] data and up to 4% for non-GPN[®]. If we reduce the training set of 10% of the data those errors become 0.5% and 6% respectively. And if we train the Markov model on only 1% of data the errors are 3.2% and 3.7% respectively.

So the global match between the Markov model and measurements is very good. We consider it validated for simulation purposes and will use it as the core of a simulation engine.

But finer predictions appear impossible from this simple model. When trying to predict a hop’s “state” from the previous ones, the model is mostly right but in a trivial way; it almost always predicts a non-spike and is thus very often right while failing to predict the rate of costly spikes themselves.

V. FUTURE MODEL COMPONENTS

Combining time-series with scaling curves allows us to extrapolate the time series. Alternatively stated, for a parametric series of time-series a linear or quadratic fit will be found; for example, a similar time-series for $b = 1, 2, 3, \dots$ may be extrapolated and simulated to a larger value of b . As per usual, the quantity of our statistical data will directly impact the quality of this simulation.

Finally, Khmelevsky’s model [32] with signal and noise function networks will be applied in future versions of the model. This could lead to less naïve “glass-box” models of the network and perhaps interact intelligently with game-state simulation.

VI. FUTURE WORK

We have gathered a fresh data set using an industry tool from WTFast that we are now looking to analyze further. This data set has more well-defined time-series information. As well, we are looking to modify this tool to create a living/continuous data collection tool that will allow for more detailed and consistent time-series data to be analyzed. By having more control over the source and destination, as well as more precise control over time-stamp and inter-packet interval, the Markov Chain analysis will have additional dimensions available to reduce false positives.

Our team has begun investigation into the use of other machine learning models such as Support Vector Models, Convolutional Neural Networks, and Reinforcement Learning. We are also looking into using unsupervised learning algorithms such as K-clustering, which may provide novel insights and classification methods we would then be able to feed back into our supervised learning models.

Another opportunity to continue our work with latency model testing is the inclusion of other game mediums. In the coming few years, WTFast hopes to offer their services for mobile, console, and cloud technologies. With these changes, we can develop experimental environments that test how these different network technologies are impacted by latency and jitter improvements, and how those technologies can be improved. While we predict that Console network conditions will prove similar to conditions seen with games played on personal computers, cloud-based and mobile-based game networks prove to have their own unique differences.

VII. CONCLUSION

When analyzing the collected data as a series of hops, each hop's delay in milliseconds represents the time it takes for a signal to reach a server and return a response. From that data, comparisons can be made between GPN[®] routes and non-GPN[®] routes. It was found that when comparing the two routes, hops in the GPN[®] route were less likely to have lengthy response times defined as spikes compared to non-GPN[®] routes. This is an early but noteworthy step in displaying the advantage of WTFast's GPN[®]. The exact reason a hop along a route may have a significantly higher response time compared to the usual can not be understood by simply collecting traceroute information like this; it could be due to a variety of factors including server configuration, hardware, etc. While the exact server-related reason can not be pinned to any specific cause with this method, the differences between GPN[®] and non-GPN[®] routes shows that the control that WTFast can have over their network compared to packets being sent down regular internet routing leads to positive results such as improved consistency.

A long-term question to investigate now is whether or not GPN[®] simulation acts more like a stock exchange graph or more like weather prediction. The former functions in a manner that is unpredictable beyond the near future or global tendencies while the latter is chaotic in nature but follows a set of general laws.

REFERENCES

- [1] B. Ward, Y. Khmelevsky, G. Hains, R. Bartlett, A. Needham, and T. Sutherland, "Gaming network delays investigation and collection of very large-scale data sets," *2017 Annual IEEE International Systems Conference (SysCon)*, pp. 1–7, 2017.
- [2] W. Doherty and A. Thadhani. (1982) The economic value of rapid response time (ibm technical report ge20-0752-0). [Online]. Available: <http://www.vm.ibm.com/devpages/jelliott/evrrt.html>
- [3] D. H. Sitrick, "Video game network. United States Patent number 4,572,509," Feb. 25, 1986.
- [4] S. G. Perlman, "Network architecture to support multiple site real-time video games. United States Patent number 5,586,257," Dec. 17, 1996.

- [5] T. Imura, H. Hazeyama, and Y. 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*, ser. NetGames '04. New York, NY, USA: ACM, 2004, pp. 116–120. [Online]. Available: <http://doi.acm.org/10.1145/1016540.1016549>
- [6] J. Jardine and D. Zappala, "A hybrid architecture for massively multiplayer online games," in *Proceedings of the 7th ACM SIGCOMM Workshop on Network and System Support for Games*, ser. NetGames '08. New York, NY, USA: ACM, 2008, pp. 60–65. [Online]. Available: <http://doi.acm.org/10.1145/1517494.1517507>
- [7] J. D. Pellegrino and C. Dovrolis, "Bandwidth requirement and state consistency in three multiplayer game architectures," in *Proceedings of the 2Nd Workshop on Network and System Support for Games*, ser. NetGames '03. New York, NY, USA: ACM, 2003, pp. 52–59. [Online]. Available: <http://doi.acm.org/10.1145/963900.963905>
- [8] P. Ghosh, K. Basu, and S. K. Das, "Improving end-to-end quality-of-service in online multi-player wireless gaming networks," *Computer Communications*, vol. 31, no. 11, pp. 2685–2698, 2008.
- [9] Q. Zhou, C. Miller, and V. Bassiliou, "First person shooter multiplayer game traffic analysis," in *Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on*, May 2008, pp. 195–200.
- [10] M. Claypool and K. Claypool, "Latency and player actions in online games," *Commun. ACM*, vol. 49, no. 11, pp. 40–45, Nov. 2006. [Online]. Available: <http://doi.acm.org/10.1145/1167838.1167860>
- [11] P. A. Branch, A. L. Cricenti, and G. J. Armitage, "An arma (1, 1) prediction model of first person shooter game traffic," in *Multimedia Signal Processing, 2008 IEEE 10th Workshop on*. IEEE, 2008, pp. 736–741.
- [12] A. L. Cricenti and P. A. Branch, "A generalised prediction model of first person shooter game traffic," in *Local Computer Networks, 2009. LCN 2009. IEEE 34th Conference on*. IEEE, 2009, pp. 213–216.
- [13] Y. Wu, H. Huang, and D. Zhang, "Traffic modeling for massive multiplayer on-line role playing game (mmorpg) in gprs access network," in *Communications, Circuits and Systems Proceedings, 2006 International Conference on*, vol. 3, June 2006, pp. 1811–1815.
- [14] B. Hariri, S. Shirmohammadi, and M. R. Pakravan, "A hierarchical HMM model for online gaming traffic patterns," in *Instrumentation and Measurement Technology Conference Proceedings, 2008. IMTC 2008. IEEE*. IEEE, 2008, pp. 2195–2200.
- [15] J. Saldana and M. Suznjevic, "Qoe and latency issues in networked games," *Handbook of Digital Games and Entertainment Technologies*, pp. 509–544, 2017.
- [16] S. W. K. Lee and R. K. C. Chang, "Enhancing the experience of multiplayer shooter games via advanced lag compensation," in *Proceedings of the 9th ACM Multimedia Systems Conference*, ser. MMSys '18. New York, NY, USA: ACM, 2018, pp. 284–293. [Online]. Available: <http://doi.acm.org/10.1145/3204949.3204971>
- [17] F. Ng, J. Hannigan, and D. Moon, "Systems and methods for managing latency in networked competitive multiplayer gaming," Apr. 25 2019, uS Patent App. 16/170,599.
- [18] D. C. Hoang, K. D. Doan, and L. T. Hoang, "Lag of legends: The effects of latency on league of legends champion abilities," 2017.
- [19] J. Saldana, "On the effectiveness of an optimization method for the traffic of tcp-based multiplayer online games," *Multimedia Tools and Applications*, vol. 75, no. 24, pp. 17 333–17 374, 2016.
- [20] J. Färber, "Traffic modelling for fast action network games," *Multimedia Tools and Applications*, vol. 23, no. 1, pp. 31–46, 2004.
- [21] T. Alstad, J. R. Dunkin, S. Detlor, B. French, H. Caswell, Z. Ouimet, and Y. Khmelevsky., "Game network traffic emulation by a custom bot," in *2015 IEEE International Systems Conference (SysCon 2015) Proceedings*, ser. 2015 IEEE International Systems Conference. IEEE Systems Council., April 13–16 2015.
- [22] A. Abdelkhalek, A. Bilas, and A. Moshovos, "Behavior and performance of interactive multi-player game servers," *Cluster Computing*, vol. 6, no. 4, pp. 355–366. [Online]. Available: <http://dx.doi.org/10.1023/A:1025718026938>
- [23] 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.
- [24] T. Jehaes, D. De Vleeschauwer, T. Coppens, B. Van Doorselaer, E. Deckers, W. Naudts, K. Spruyt, and R. Smets, "Access network delay in networked games," in *Proceedings of the 2nd workshop on Network and system support for games*. ACM, 2003, pp. 63–71.

- [25] Y. Gao, L. Wang, Z. Xie, W. Guo, and J. Zhou, "Energy-efficient and quality of experience-aware resource provisioning for massively multiplayer online games in the cloud," in *International Conference on Service-Oriented Computing*. Springer, 2018, pp. 854–869.
- [26] V. Burger, J. F. Pajo, O. R. Sanchez, M. Seufert, C. Schwartz, F. Wamser, F. Davoli, and P. Tran-Gia, "Load dynamics of a multiplayer online battle arena and simulative assessment of edge server placements," in *Proceedings of the 7th International Conference on Multimedia Systems*. ACM, 2016, p. 17.
- [27] S. A. Abdulazeez and A. El Rhalibi, "Dynamic load balancing for massively multiplayer online games using opnet," in *International Conference on E-Learning and Games*. Springer, 2018, pp. 177–191.
- [28] M. M. Garcia and R. M. Woundy, "Network latency optimization," Jun. 12 2018, uS Patent 9,998,383.
- [29] S. A. Abdulazeez, A. El Rhalibi, and D. Al-Jumeily, "Simulation of massively multiplayer online games communication using opnet custom application," in *2016 IEEE Symposium on Computers and Communication (ISCC)*. IEEE, 2016, pp. 97–102.
- [30] J. N. Plumb, S. K. Katera, and R. Stutsman, "Hybrid network clusters using common gameplay for massively multiplayer online games," in *Proceedings of the 13th International Conference on the Foundations of Digital Games*, ser. FDG '18. New York, NY, USA: ACM, 2018, pp. 2:1–2:10. [Online]. Available: <http://doi.acm.org/10.1145/3235765.3235785>
- [31] I. Slivar, L. Skorin-Kapov, and M. Suznjevic, "Qoe-aware resource allocation for multiple cloud gaming users sharing a bottleneck link," in *2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*. IEEE, 2019, pp. 118–123.
- [32] Y. Khmelevsky, H. Mahasneh, and G. J. Hains, "A stochastic gamer's model for on-line games," in *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, 2017, pp. 1–4.
- [33] J. P. McManus, T. G. Day, and Z. J. Mailloux, "The effects of latency, bandwidth, and packet loss on cloud-based gaming services," 2019.
- [34] Y. Chen, J. Liu, and Y. Cui, "Inter-player delay optimization in multiplayer cloud gaming," in *2016 IEEE 9th International Conference on Cloud Computing (CLOUD)*. IEEE, 2016, pp. 702–709.
- [35] W. Cai, R. Shea, C.-Y. Huang, K.-T. Chen, J. Liu, V. C. Leung, and C.-H. Hsu, "A survey on cloud gaming: Future of computer games," *IEEE Access*, vol. 4, pp. 7605–7620, 2016.
- [36] S. Mallick, G. Hains, and C. S. Deme, "An alert prediction model for cloud infrastructure monitoring," 2013.
- [37] S. M. Ross, *Introduction to probability models*. Academic press, 2014.
- [38] G. Hains, Y. Khmelevsky, R. Bartlett, and A. Needham, "Game private networks performance: From geolocation to latency to user experience," in *2017 Annual IEEE International Systems Conference (SysCon)*, April 2017, pp. 1–6.
- [39] T. Alstad, R. Dunkin, 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," 07 2015.
- [40] G. Hains, Y. Khmelevsky, R. Bartlett, and A. Needham, "Game private networks performance: analytical models for very-large scale simulation," in *2016 IEEE International Conference on Cybercrime and Computer Forensic (ICCCF)*, June 2016, pp. 1–10.
- [41] S. M. Ross, *Introduction to probability and statistics for engineers and scientists*. Academic press, 2004.
- [42] M. AB. Minecraft home page. [Online]. Available: <https://minecraft.net/>
- [43] COSC 470 SW Engineering Capstone Project Course Team, "A short video clip with 50 bots running in a square." Computer Science Department, Okanagan College. [Online]. Available: <https://www.youtube.com/watch?v=KYrIO7yWekw>
- [44] Gamepedia. Infiniminer. [Online]. Available: <http://tinyurl.com/o5plsbk>
- [45] GitHub, Inc. DarkStorm652/DarkBot. Minecraft thin client and automation framework. [Online]. Available: <https://github.com/DarkStorm652/DarkBot>
- [46] GitHub Inc., Steveice10/MCProtocolLib. A library for communications with a minecraft client/server. [Online]. Available: <https://github.com/Steveice10/MCProtocolLib>
- [47] C. Limited. Memyadmin 2 the minecraft control panel. [Online]. Available: <https://www.mcmyadmin.com>.
- [48] Wikipedia. Minecraft. [Online]. Available: <http://en.wikipedia.org/wiki/Minecraft>
- [49] Wikipedia. (2014) Minecraft. [Online]. Available: <http://en.wikipedia.org/wiki/Minecraft>
- [50] K. Yang, X. Zhang, Y. Zhao, Q. Fan, Q. Gao, Y. Lyu, H. Yin, and Z. Ma, "Looking into online gaming from measurement perspective," in *2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)*. IEEE, 2019, pp. 203–2035.
- [51] S. E. Middleton and S. Modafferi, "Scalable classification of qos for real-time interactive applications from ip traffic measurements," *Computer networks*, vol. 107, pp. 121–132, 2016.
- [52] R. Saunders, J. Cho, A. Banerjee, F. Rocha, and J. Van der Merwe, "P2p offloading in mobile networks using sdn," in *Proceedings of the Symposium on SDN Research*. ACM, 2016, p. 3.
- [53] P. Moll, M. Lux, S. Theuermann, and H. Hellwagner, "A network traffic and player movement model to improve networking for competitive online games," in *2018 16th Annual Workshop on Network and Systems Support for Games (NetGames)*. IEEE, 2018, pp. 1–6.
- [54] A. Sackl, R. Schatz, T. Hossfeld, F. Metzger, D. Lister, and R. Irmer, "Qoe management made uneasy: The case of cloud gaming," in *2016 IEEE International Conference on Communications Workshops (ICC)*. IEEE, 2016, pp. 492–497.