Game Private Networks Performance: from geolocation to latency to user experience

Gaétan Hains*
Huawei, France Research Centre, Paris
Email: gaetan.hains@huawei.com
*LACL, Univ. Paris-Est Créteil, France

Youry Khmelevsky†
Computer Science, Okanagan College, Canada
Email: ykhmelevsky@okanagan.bc.ca
†Computer Science, UBC Okanagan, Canada

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

***Abstract*—Online games are interactive competitions by players who compete in a virtual environment. The WTFast's Gamers Private Network (GPN®) is a client/server solution that makes online games faster. It connects online video-game players with a common game service across a wide-area network. Response time, latency and its predictability are keys to GPN® success and runs against the vast complexity of internet-wide systems.**

We have built an experimental network of virtualized GPN® components so as to carefully measure the statistics of latency for distributed Minecraft games and to do so in a controlled laboratory environment. This has led to a better understanding of the coupling between parameters such as: the number of players, the subset of players that are idle or active, the volume of packets exchanged, the size of packets, latency to and from the game servers, and time-series for most of those parameters.

In a new experiment described here, we use traceroute measurements and IP-address geolocations to quantify the intuitive correlation in wide-area GPN® that correlates longer message routes to degraded game experience. Our conclusions are related to connection types for different game styles and validate technologies that aim at reducing latency variability through a better control of message routing. This is another small step towards rational, quantified and service-oriented minimization/stabilization of GPN latency by relating explicitly the choice of connection routes with game experience.

I. Introduction

The "GPNPerf" (2014-2015) project has built a laboratory version of a Games Private Network ® that is used for extensive and controlled-environment experiments to investigate the conditions of low and stable latency in online games. Experiments conducted since 2014 with the Minecraft network game have produced an ever-increasing quantity, quality and variety of measurements. This will allow us to simulate very large game configurations at internet scale with moderate computational resources by using Markov models of message transport.

In this paper we describe a complementary experiment that measured individual traceroutes across the internet and correlated their overall delay with the number of hops, and the source- to target geographical locations. This allows us to quantify for the first time the choice of internet route from game player to server with the actual game user experience. Services such as WTFast's Gamers Private Network (GPN®) take advantage of shorter routes through internet and their final-user value has thus been quantified for the first time. This research creates new practically-oriented public knowledge, and yet allow proprietary technologies to dynamically improve their message routing for specific game objectives in specific geolocations of the players and servers.

II. Existing works

Predictable and sub-second response time has long been a key concern for interactive computer systems [1]. For a majority of (local) video games this is a requirement that modern hardware has satisfied, despite a continuous rise in graphics and interaction requirements. A *video game network* is a *distributed* set of "apparatus which are capable of exhibiting an interactive single identity game", as defined in a patent dated 1986 [2]. 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" [3].

The last decade had seen a growing interest in research about this problem. Some researchers like Iimura, Jardine and co-authors have proposed peer-to-peer architectures for multiplayer online video games [4], [5], with the intention of reducing the bandwidth and processing requirements on servers. Pellegrino et al. [6] have then proposed a hybrid architecture called P2P with central arbiter. The bandwidth requirements on the arbiter are lower than the server of a centralized architecture.

Some authors discuss interactive online games, especially ones related to the "first person shooter (FPS)" [7], [8] and network traffic for such games [9]. Latency is a challenge for online games, as reported in [10], [6] and [11] and it's an important factor of an online gaming experience.

Zhou, Miller, and Bassilious [9] have made the obvious but central observation that "Internet delay is important for FPS (first-person shooter) games because it can determine who wins or loses a game." Many specific game interactions are time sensitive, but it is the time the information reaches the server that matters, not the time the player actually pushes the button.

Claypool and Claypool [10] have observed that Internet latency's effect is strongest for games with a first-person perspective and a changing model like for example Minecraft that was used in our earlier laboratory-controlled experiments.

More recent studies [7], [8] of first-person shooter games have modelled time series behaviour of game traffic and tested the model on up to eight different games. Indeed the study of

978-1-5090-4623-2/17/$31.00 ©2017 IEEE

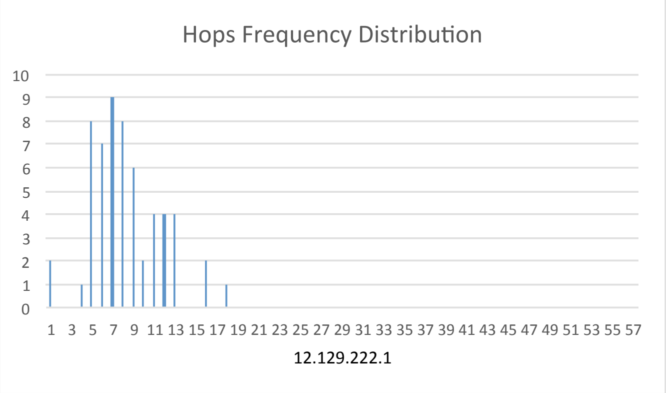


Fig. III.1. Frequency distribution of number of hops in a route

Wu, Huang and Zhang [12] shows that "the server-generated traffic has a tight relationship with specific game design", again from our point of view confirming the need for precise measurements of a given network on a single game.

III. IP-addresses, geolocation and distance

The first step in our study has been to compile and analyze many hundreds of pairs of distant IP addresses for traceroute delay, both inside a GPN (provided by WTFast) and through normal internet paths. All data entries consist of a game client (source) IP address and a game server (target) IP address, together with `traceroute` data in the form of a variable-length sequence of delays in ms units. The length of the sequence is called *number of "hops"* i.e. relays in the client-to-server paquet route. The sum of its delays is the total ping time that approximates the latency a game player experiences when interacting with the game server.

Several statistics can be observed like the frequency distribution of the paquet route lengths to a given server. In the example shown in figure III.1 this variable has a right-skewed bell-shaped distribution around the value of 7 hops and mostly spreads from 5 to 13 on a sample of about 50 traceroutes.

As anyone would expect the number of hops is directly proportional to total ping time. But the experiments reveal much more information about this relationship as seen in figure III.2. The total ping time is **linearly** correlated with the number of hops and the actual ratio is approximated to about 20% by

$$\text{Ping time (ms)} \approx 16 \times \text{Number of Hops}.$$

It is interesting to compare our measured distribution of hops and their effect on ping time with a relatively old but careful and systematic study by Obraczka and Silva. Their measurements (tables IV and V in [13]) show numbers of hops that average from 10 to 15, sensibly more than what is seen in our figure III.1. Yet their estimated total latency per number of hops shows values comparable to our factor of 16 mentionned above: the interpolated lines in Figure 2 "RTT versus hops" of [13] have slopes varying from about 12 to not much more than 20.

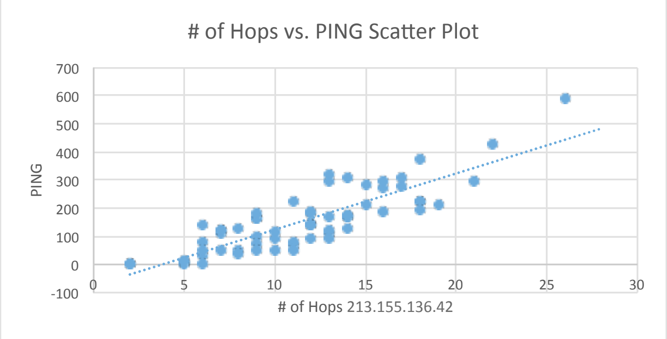
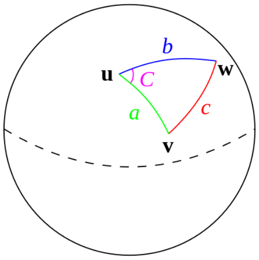


Fig. III.2. Ping delay vs number of relays in a route $y \approx 20.8\ x - 80$

$$\mathbf{hav}(c) = \mathbf{hav}(a - b) + \sin(a)\sin(b)\ \mathbf{hav}(C).$$ [9]



Spherical triangle solved by the law of haversines.

Fig. IV.1. Haversine formula

IV. Traceroutes, number of relays

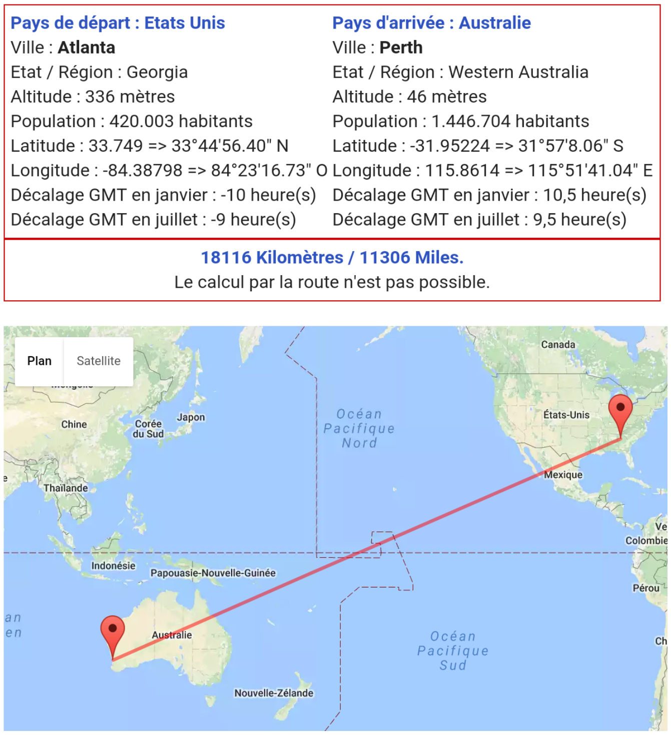
We used an online geolocation service [14] to discover each IP address' geographical coordinates in the form of a latitude (^oE) and a longitude (^oN). Each one of our tables refers to a fixed server and a variable number of clients sending queries to it. For each entry i.e. each traceroute measurement from a client to the given server, we computed [15] the client-server geographical distance by the Haversine formula [16]

$$\mathtt{hav}(c) = \mathtt{hav}(a - b) + \sin(a)\sin(b)\mathtt{hav}(C)$$

that determines the distance between two points on a sphere (figure IV.1) .

Part of the resulting tables is shown in figure IV.3.

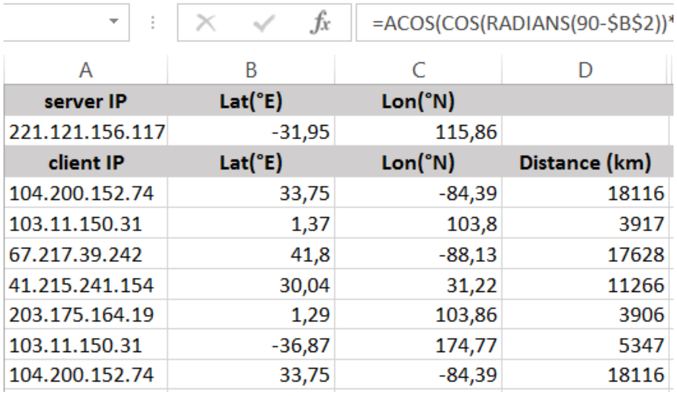
Then we looked for linear relationships among the input variables (distance, number of hops, ping time): how many μs of delay per km of distance, how many ms per hops in the route, how many km of distance travelled per hop segment and how many hops per 1000km of distance to cross IV.4. One typical such dataset shows an average of 570ns/km of distance, 479μs/hop to cross and 1295 km travelled per hop. The average deviations of those ratios go from 40 to 60%. So in simple terms, there are linear relationships between distance, hops and delay that can be grossly approximated by a linear ratio with 50% error (figure IV.5).



Pays de départ : Etats Unis	**Pays d'arrivée : Australie**
Ville : **Atlanta**	Ville : **Perth**
Etat / Région : Georgia	Etat / Région : Western Australia
Altitude : 336 mètres	Altitude : 46 mètres
Population : 420.003 habitants	Population : 1.446.704 habitants
Latitude : 33.749 => 33°44'56.40" N	Latitude : -31.95224 => 31°57'8.06" S
Longitude : -84.38798 => 84°23'16.73" O	Longitude : 115.8614 => 115°51'41.04" E
Décalage GMT en janvier : -10 heure(s)	Décalage GMT en janvier : 10,5 heure(s)
Décalage GMT en juillet : -9 heure(s)	Décalage GMT en juillet : 9,5 heure(s)

18116 Kilomètres / 11306 Miles.
Le calcul par la route n'est pas possible.

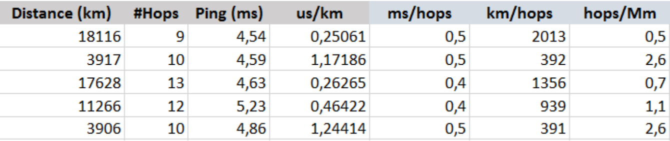
Fig. IV.2. Geographical distance computed from geolocation



=ACOS(COS(RADIANS(90-B2))*

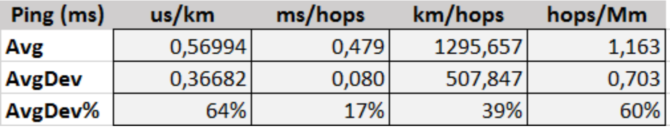
A	B	C	D
server IP	**Lat(°E)**	**Lon(°N)**	
221.121.156.117	-31,95	115,86	
client IP	**Lat(°E)**	**Lon(°N)**	**Distance (km)**
104.200.152.74	33,75	-84,39	18116
103.11.150.31	1,37	103,8	3917
67.217.39.242	41,8	-88,13	17628
41.215.241.154	30,04	31,22	11266
203.175.164.19	1,29	103,86	3906
103.11.150.31	-36,87	174,77	5347
104.200.152.74	33,75	-84,39	18116

Fig. IV.3. Distance computed field



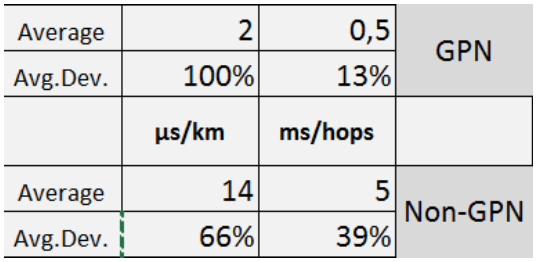
Distance (km)	#Hops	Ping (ms)	us/km	ms/hops	km/hops	hops/Mm
18116	9	4,54	0,25061	0,5	2013	0,5
3917	10	4,59	1,17186	0,5	392	2,6
17628	13	4,63	0,26265	0,4	1356	0,7
11266	12	5,23	0,46422	0,4	939	1,1
3906	10	4,86	1,24414	0,5	391	2,6

Fig. IV.4. Ratios between the input variables



Ping (ms)	us/km	ms/hops	km/hops	hops/Mm
Avg	0,56994	0,479	1295,657	1,163
AvgDev	0,36682	0,080	507,847	0,703
AvgDev%	64%	17%	39%	60%

Fig. IV.5. Averages of ratios between the input variables



Average	2	0,5	GPN
Avg.Dev.	100%	13%	
	µs/km	**ms/hops**	
Average	14	5	Non-GPN
Avg.Dev.	66%	39%	

Fig. V.1. Effect of GPN on ping times

V. The effect of the GPN

The measurements and calculations summarized above come from paquet routes that were selected by WTFast's GPN routing. This technology stabilizes ping latency by decreasing the variability of the number of hops, choosing better geographical routes and eliminating large-delay hops that occur randomly within internet routes.

To verify its positive effect on latency and latency variance we have compared the statistics of GPN-routed with non-GPN (i.e. freely routed internet) `traceroute` requests. To measure this difference we took a sample of 20 routes through the GPN and 29 non-GPN (normal internet) routes. The individual client-server pairs are different so this measurement only measures group behaviour. But the average and variance effect is striking with the GPN providing an acceleration factor of 7 for μs/km of distance and an acceleration factor of 10 for ms/hops (Figure V.1). That speedup factor of 7x is observed on a given set of source-destination pairs and is decisively in favour of GPN. The dataset of section V for which non-internet latency was much lower (570ns/km) as it comes from different routes and distances. Only the linear correlations and relative factors appear reliable in our current measurements, future data analysis may reveal hidden variables to explain this variability. This measurement is clear evidence that GPN drastically reduces latency on average but a key factor of game experience quality is also that latency should have a low variance (or average deviation). In our measurement the delay per hop for GPN routing is 0.5ms/hop $+/-13\%$ against 5ms/hop $+/-39\%$ for non-GPN routing. That is a reduction of latency variance by a factor of 3 which is remarkable. But, when comparing variance for the latency *by kilometer travelled* this effect is not visible: the GPN routes exhibit 2μs/km $+/-100\%$, a large variance while the non-GPN routes show a value of 14μs/km $+/-66\%$. In absolute values that is an important reduction of latency variance but in relative terms it is not.

A last confirmation of the positive effect of GPN technology is the elimination of ”freak” hop times i.e. delays in the traceroute time series that vary dramatically from the average. None of those has been observed in our sample for GPN routes. But many non-GPN routes include such delays that incur for most of the relative slowdown factor we measured for averages. An example is shown in figure V.2. All hop delays last 10ms or less except for an extremely high one at

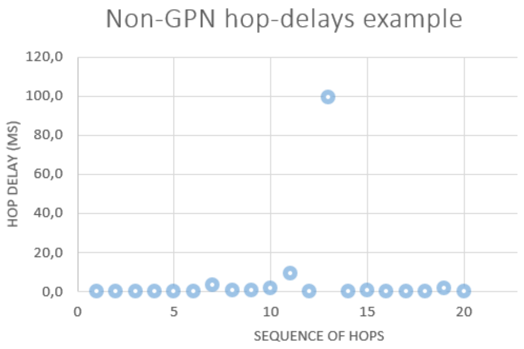


Fig. V.2. Example of non-GPN route: sequence of hop delays

100ms. Such a wait time of several hundreds of ms might be acceptable for a strategy game but can seriously degrade de user experience for a FPS or other action game. As a result we can conclude that GPN is an important and decisive factor in preserving the quality of FPS games, by reducing ping latency and its variance by a factor of 7 to 10. More experiments and analyses will be conducted to confirm this over very large datasets but the coherence and clarity of our current observations is a strong sign in favour of the lasting value of GPN technology for massively online games.

VI. Large dataset

To observe finer tendencies in the key variables and move closer to statistical confirmation of our initial observations we have gathered and analyzed a dataset of 30 000 traceroute records. Its measured- and computed fields are the same as before: origin IP, target IP (game server), latitude-longitude of both, geodistance, list of hop-delays, total ping time, number of hops. Among other statistics we have computed and plotted total ping delay (ms) vs client-server geodistance (km) for GPN routes.

The cloud of measurements does show a "hole" between 5000km and 7000km that is explained by the fact that most routes leave North-America and either lead to a server located in North-America or Eurasia. The hole thus corresponds either to the Pacific or the Atlantic ocean, having understood that there were no target IP addresses in South-America.

Normal internet (non-GPN) latencies are approximately linear with geodistance, much lower than GPN latencies and again they exhibit a hole between 5000 and 7000km that is an artefact of our datasets.

Unfortunately, systematic but unidentified errors in the very large datasets produce unrealistic values for signal transport so cannot yet publish their details yet. The only conclusions are that the general trends and relative values of GPN- versus non-GPN measurements appear to be confirmed in the large scale.

VII. Large-scale simulation

Each sequence of traceroute hop delays is a time-series of delays that can be simulated by a Markov chain as in [18]. The states of the stochastic process can be intervals for the value of a certain hop delay in ms, for example 0-1ms, 1-2ms ... up to a maximal value that is no more than 10 in our examples. The state change probabilities are estimated by the relative frequencies of changing from a given state to another one in the sample of all successive values of the time series [19]. An initial probability distribution of hop delays can similarly be estimated by absolute frequencies of the delay values in the time series: if all delays were of 5ms then the state 4-5ms would be estimated to 100% probability and all others 0% for example. Once an initial probability vector and Markov transition matrix have been computed from a sample, it becomes possible to simulate that set of routes in the GPN or non-GPN networking: successive multiplications of the Markov matrix by the initial probability distribution yield successive distribution vectors for the likely hop delays. One 10x10 matrix is multiplied by a 10-vector of floating-point values in about 200 Flops and since most routes contain less than 20 hops, this simulation can complete in less than 4000 Flops. As a result a traceroute can be given a realistic simulation in a few thousand Flops i.e. less than a microsecond on modern architectures.

The traceroute delays we have measured had often total ping times of 5-10ms, so their simulation in less than a μs is an acceleration factor of more than 5000. This means that, given as many cores as simulated game players, the latency experienced by each one can be simulated more than 5000 faster than real-time experiments in internet or GPN. We also have designed similar models for game-server response times. Both models will be integrated into our construction of a large-scale simulation of game networks at internet scale, allowing us to explore specific network and game situations beyond that are either too large or impossible to replicate in real networks.

VIII. Conclusion

The GPNPerf project aims at a deep understanding of game network latency. It appears natural not to consider bandwidth saturation effects for online games are not as heavy in traffic consumption as video streaming, data streaming for example. In all our experiments the amount of traffic was very low compared with the overall bandwidth of the network equipment. We consider that game traffic volume is generally negligible with respect to network bandwidth so should have no visible effect in practical situations.

In this work we have measured and analyzed the effect of GPN on reducing latency and latency variance between client and game server. By normalizing ping latency to geo-located distance in km, and also to the number of hops in each traceroute we have identified a speed-up factor of up to 10x in favour of GPN against normal internet routing. The variance, hence predictability of ping latency is also reduced by an important factor under the same conditions.

The geolocation web service that we used [14] is based on a database of IP addresses corresponding to cities. But such databases are know to be imperfectly reliable. For example Poese et al. [20] conclude to *country-level accuracy, but* *certainly not city-level* accuracy for geolocation databases. The reader is therefore warned that our exact statistics remain to be confirmed on more varied datasets.

Then we have confirmed and refined the major trends by analyzing a larger dataset of 30 000 measurements. Some of its features have to be confirmed like the small but non-empty set of high-latency points for GPN. The exact features of non-GPN traceroutes, like their unique high-latency hops must also be explained statistically or even deterministically. Larger sets of non-GPN points should be explored and, our main missing measurement is the comparison of (large numbers of) identical client-server routes. Our current observations compare all variables for their statistics and distributions, but only a few compare the same route for GPN and non-GPN. None of the measures implies that this will not confirm our current analysis but that remains to be completely verified.

Finally we have described the mathematical elements and computational speed of a simulator module for this key factor of online game experience, namely the sequence of delays that messages incur from player to game server.

Future and ongoing complementary work will confirm this initial analysis, refine it with time-of-day, calendar or game condition parameters so as to produce a realistic laboratory simulator of very-large scale game networks. This simulator will then be used to cover extreme conditions that may occur in internet or GPN, and design dynamic solutions to minimize their effect on game-player experience.

Acknowledgment

The research project results described in this paper were achieved with support from Computer Science department at Okanagan College and by 2 NSERC grants: GPNPerf — Investigating performance of game private networks in 2014 (CCI ARD1 465659 - 14), and GPNPerf2 — Game private networks and game servers performance optimization in 2016 (CCI ARD2 477506-14).

GPN traceroute data was provided by WTFast. Okanagan College students measured non-GPN traceroute delays and compiled all raw data in this study.

References

[1] 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

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

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

[4] T. Iimura, 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

[5] 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

[6] 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

[7] 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.

[8] 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.

[9] Q. Zhou, C. Miller, and V. Bassilious, “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] 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.

[12] 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.

[13] K. Obraczka and F. Silva, “Network latency metrics for server proximity,” in *Global Telecommunications Conference, 2000. GLOBECOM ’00. IEEE*, vol. 1, 2000, pp. 421–427 vol.1.

[14] 2016, freeGeoIP geolocation web service www.freegeoip.net.

[15] Rosettacode.org, “Rosettacode.org: Haversine formula,” 2016, http://rosettacode.org/wiki/Haversine_formula#Ocaml.

[16] Wikipedia, “The Haversine formula,” 2016, https://en.wikipedia.org/wiki/Haversine_formula.

[17] A. Singla, B. Chandrasekaran, P. B. Godfrey, and B. Maggs, “The internet at the speed of light,” in *Proceedings of the 13th ACM Workshop on Hot Topics in Networks*, ser. HotNets-XIII. ACM, 2014, pp. 1:1–1:7.

[18] S. Mallick, G. Hains, and C. S. Deme, “An alert prediction model for cloud infrastructure monitoring,” 2013.

[19] S. M. Ross, *Introduction to probability models*. Academic press, 2014.

[20] I. Poese, S. Uhlig, M. A. Kaafar, B. Donnet, and B. Gueye, “IP geolocation databases: Unreliable?” *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 2, pp. 53–56, 2011.

[21] S. M. Ross, *Introduction to probability and statistics for engineers and scientists*. Academic press, 2004.

[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] M. AB. Minecraft home page. [Online]. Available: https://minecraft.net/

[24] 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.

[25] 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.

[26] 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

[27] J. Färber, “Traffic modelling for fast action network games,” *Multimedia Tools and Applications*, vol. 23, no. 1, pp. 31–46, 2004.

[28] Gamepedia. Infiniminer. [Online]. Available: http://tinyurl.com/o5plsbk

[29] GitHub, Inc. DarkStorm652/DarkBot. Minecraft thin client and automation framework. [Online]. Available: https://github.com/DarkStorm652/DarkBot

[30] GitHub Inc., Steveice10/MCProtocolLib. A library for communications with a minecraft client/server. [Online]. Available: https://github.com/Steveice10/MCProtocolLib

[31] C. Limited. Mcmyadmin 2 the minecraft control panel. [Online]. Available: https://www.mcmyadmin.com.

[32] Wikipedia. Minecraft. [Online]. Available: http://en.wikipedia.org/wiki/Minecraft

[33] 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.

[34] 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.

[35] Wikipedia. (2014) Minecraft. [Online]. Available: http://en.wikipedia.org/wiki/Minecraft