Generating and Visualizing Message Latency Metrics
(Technical Paper)

Measuring Latency to Improve Online Game Experience

(STS Paper)

A Thesis Prospectus Submitted to the

Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements of the Degree
Bachelor of Science, School of Engineering

Daniel Zhao
Fall, 2022

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Signature _____ Date _____
Daniel Zhao

Approved _____ Date _____
Briana Morrison, Department of Computer Science

Approved _____ Date _____
Richard D. Jacques, Department of Engineering and Society

Introduction

The ARPANET email system, created in 1971, was the first example of a distributed system – a process that exists a network of nodes (e.g., computers) (Gibbs 2016). Since then, many new technologies have utilized the parallel computing power and scalability that distributed systems provide yet are faced with a central problem: latency within network nodes.

Latency is the time taken for a message to transmit between any two nodes of a network. Measuring and analyzing latency is incredibly important for network health, as extremely high latency is indicative of a failure of a node or server, and "results in poor performance, negatively affects SEO, and can induce users to leave the … application altogether" (Cloudflare). High network latency will be explored in two scenarios throughout this prospectus: chat message latency, specifically the Wickr Pro app, and videogame ping, specifically in Among Us.

Wickr is a business-oriented, secure end-to-end encrypted messaging platform that was recently purchased by Amazon Inc (Wickr Io). Since Wickr a relatively novel chat system, it that lacks a tool to measure and report on message latency. Furthermore, Wickr contains an enterprise version called Wickr Pro that contains an API library to create automated workflows. Because of this, Wickr Pro is perfect to generate and visualize latency metric data. As such, my technical topic will be to design and implement a bot to intermittently send chat messages to scrap latency values and create a visualization platform for which these latency summaries can be viewed and acted upon. This bot's purpose is to notify engineers when latency spikes to reduce system downtime and mitigate issues proactively.

Among Us is a popular video that gained popularity in 2020 at the height of the Coronavirus pandemic. The appeal was likely due to the social nature of the game during a time when face-to-face interactions were limited (Lorenz 2020). However, Among Us suffers from

extremely high network latency, which negatively affects player satisfaction and leads to player dissatisfaction and leaving games prematurely (Chen 2008). For my STS topic, I will focus on the impact of network latency on the experience and psychology of videogame players.

Generating and Visualizing Message Latency Metrics

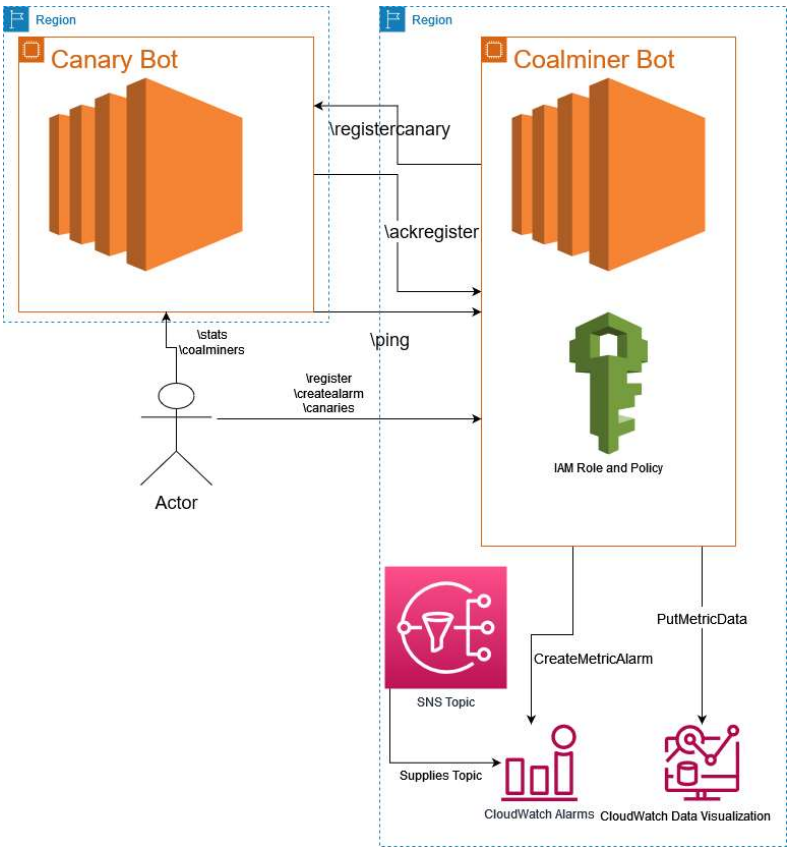
The Wickr Pro chat system is a relatively new chat system that incorporates secure end-to-end encryption and business-level administrative controls, as well as a backend API to create rooms, send messages, etc. (Wickr Io). However, as a newer chat system, Wickr Pro engineers were reactively, rather than proactively, addressing issues that could cause a downtime in service. I observed firsthand how many components lacked a metric collection process, causing the system overall to not perform efficiently because of unnoticed flaws.

Custom integrations can be developed with WickrIO, the API library provided by Wickr Pro. The integrations are built on NodeJS and hosted on custom Docker images. All integrations consist of creating bot users through an admin dashboard, loading the necessary code onto the integration, and then interacting with the bot using commands in any chatroom the bot resides in. Although the WickrIO system is proprietary to Wickr, it still allows integrations with third party software, such as AWS compatibility, and thus offers powerful capabilities. Prior to my project, the Wickr DevOps team began creating a pair of bots that could expose the message latency using WickrIO. The goal was to create an internal tool that existed on the Wickr client to measure and send data to Prometheus, a data visualization tool. The project was able to produce a working bot-pair, however the team quickly discovered a host of developmental issues and dependency problems. More precisely, due to miscommunication between the DevOps and Bots teams, this early version of the Canary Bot project utilized a legacy API for the bots'

development. The bot would periodically crash and eventually stop sending data. As such, the first attempt at message diagnostics was shelved.

For my technical project, I iterated on the previous design and implemented a pair of bots to continuously send ping messages and measure the delay in send and receive times. Though it had many problems, this first bot was well designed and well architected, and I ended up designing my solution based on this previous iteration. I changed a few key components to better utilize the native AWS support and provide engineers more diagnostic data. Firstly, because Wickr is hosted entirely on AWS Cloud infrastructure, I architected all components to use native AWS components as it was more sensible and secure. Secondly, I could improve the utility and scalability of the bot by updating to the most recent API version and designing with a factory design pattern. And lastly, I integrated the bots with native dashboard visualization and metric alarming to decrease response time of engineers in critical events. The final design consisted of two bots communicating with each other, one to send messages and another to process the messages and create easily visualizable data points and graphs.

Canary-Coalminer Bot Architecture



Two bots exist in two separate regions and communicated through interval pings. Pings are then processed by the Coalminer and sent to CloudWatch for metric analysis and alarming. Users can interact with the bot pair through chat commands to expose relevant statistics. (Created by Author)

Measuring Latency to Improve Online Game Experience

In the past few decades, gaming has become an important part of pop culture, entertainment, and industry. According to Statista, the gaming industry has earned $200 billion dollars in revenue in 2022, not including all revenue earned in eSports, which has become a billion-dollar industry, and streaming platforms such as Twitch and YouTube (Statista). However, with the popularity of online games, scalability poses a huge issue, especially across

international borders. According to Chen, network lag severely hinders a player's experience, and "affects a players decision to leave a game prematurely" (Chen 2008). Many games require little to no latency to deliver a crisp and satisfying experience to players. In shooter games, as much as 100 millisecond delay can have drastic effects on the outcome of a situation. Thus, the necessity of low latency is exacerbated in competitive online gaming.

Among Us particularly suffers from network latency issues. As mentioned above, Among Us rose in popularity during the height of the COVID-19 pandemic (Lorenz 2020). Although the game had been released for two years, the indie game did not expect such a sudden and large player base – as much as 3 million concurrent players – and so regional servers were constantly experiencing lag or failing (King 2020). This meant game synchronization issues between players (i.e., players being shown different events), players being forcibly disconnected from games, or game servers failing altogether. For many people, online gaming such as Among Us provided necessary social interaction in an isolated age, and without stable networks, this quality time would be hindered.

In the same way that measuring and displaying latency benefits distributed systems such as the Wickr app, exposing and acting upon these metrics in games such as Among Us can greatly benefit the system and users. By itself, a catch-all latency metric provides little value. However, latency metrics that subdivide the components of the network transmission would be far more beneficial to engineers to diagnose issues. Subdividing the latency into different components such as Network Delay, Processing Delay, and Playout Delay (Chen 2008) would facilitate scalable development by informing developers the root cause of the high latency. Once informed, developers can then attempt to improve the latency through automatically balancing latency, employing path inlining, outlining, or cloning (Zander 2005, Mosberger 1996).

This network latency measurement in gaming is similarly tied to my technical project, which seeks to measure and display latency within chat message systems for diagnostic usage. By researching and investing more into quick and reliable communication networks, distributed technologies could be greatly improved to run more efficiently, which benefits every internet-related industry.

Conclusion

A pair of bots was created to generate latency metrics for Wickr Pro with a goal of improving network performance. The bot sent messages and scraped latency data from the message transmission, uploaded the data to AWS CloudWatch, and created a visualization dashboard. This solution provided engineers with percentile summary data related to the transmission time of chat messages, from which the engineers can analyze and act upon. This technology can be expanded to the upcoming industry of online gaming, which suffers from large network latencies due to the global distribution of players. Among Us is a perfect case study for network latency, as it spans across many regions and thus suffers from poor synchronization. Exploring how this technology can benefit Among Us can also benefit many other online games, and show the potentials and limitations of measuring latency to improve game experience.

References

Kwon, M. (2015). A Tutorial on Network Latency and Its Measurements. In T. Soyata (Eds.), Enabling Real-Time Mobile Cloud Computing through Emerging Technologies (pp. 272-293). IGI Global. https://doi.org/10.4018/978-1-4666-8662-5.ch009

Gibbs, S. (2016, March 7). How did email grow from messages between academics to a global epidemic? Retrieved October 19, 2022, from https://www.theguardian.com/technology/2016/mar/07/email-ray-tomlinson-history

Cloudfare. (n.d.). *What is latency? | how to fix latency | cloudflare*. Cloudfare.com. Retrieved October 25, 2022, from https://www.cloudflare.com/learning/performance/glossary/what-is-latency/

Lorenz, T. (2020, October 14). Everyone's Playing Among Us. The New York Times. Retrieved October 25, 2022, from https://www.nytimes.com/2020/10/14/style/among-us.html

Chen, K.-T. `, Huang, P., & Lei, C.-L. (2008, August 8). Effect of network quality on player departure behavior in online games. IEEE Xplore. Retrieved October 25, 2022, from https://ieeexplore.ieee.org/abstract/document/4591393

Wickr Io. Wickr IO Overview. (n.d.). Retrieved September 30, 2022, from https://wickrinc.github.io/wickrio-docs/#wickr-io

Video games - worldwide: Statista market forecast. Statista. (n.d.). Retrieved October 25, 2022, from https://www.statista.com/outlook/dmo/digital-media/video-games/worldwide#revenue

King, A. (2020, October 8). Why among US servers keep going down. ScreenRant. Retrieved October 26, 2022, from https://screenrant.com/among-us-server-down-issues-problems-ping-timeout/

Zander, S., Leeder, I., Armitage, G., & Metrics, O. M. V. A. (2005, June 15). *Achieving fairness in multiplayer network games through automated latency balancing*. ACM Other conferences. Retrieved October 26, 2022, from https://dl.acm.org/doi/abs/10.1145/1178477.1178493?casa_token=ONIDUPZz_IoAAAAA%3Ag6dbxlhWJK6_pBkweH8HNb_Pa2bPqzxgnKJakw31eaGxHUUPKCPgl3QjiTdMSjxViqhmLgK5i1ZD

Mosberger, D., Peterson, L. L., Bridges, P. G., & O'Malley, S. (1996, October 1). *Analysis of techniques to improve protocol processing latency*. ACM SIGCOMM Computer Communication Review. Retrieved October 26, 2022, from https://dl.acm.org/doi/abs/10.1145/248157.248164