TECHNICAL REPORT

COMP3670

COMPUTER NETWORKS

Types of Networked Games and Performance Issues Related to Online Gaming

Author:
Dennis Dao, SID:110010116

Submitted to:
Dr. Ikjot Saini

July 25, 2022

Types of Networked Games and Performance Issues Related to Online Gaming

Dennis Dao
School of Computer Science
University of Windsor
Windsor, ON, Canada
dao114@uwindsor.ca

***Abstract*—Video games are electronic games played with a controller and give visual feedback to its end-users. Multiplayer games are games played by many players - whether they are human players or the game itself using artificial intelligence (AI). These games can be played within a local network or online over a large geographic area up to hundreds or thousands of players. Games can also be played on the cloud, where single players only need a subscription to the game hosted online. These are examples of networked games, where computers connect to each other during an online session. There are types of networked games and they can be identified by their architecture. This and their overall performance issues are analyzed. A discussion is written on the thoughts about each type, performance issues with online gaming, the work that has been done in the past, and what could be done in the future.**

***Index Terms*—Client-server, peer-to-peer (P2P), massively multiplayer online games (MMOG), latency, scalability, consistency, bandwidth, throughput, fault tolerance, UDP, TCP, Quality of Experience (QoE), local area network (LAN), wide area network (WAN)**

I. Introduction

The Internet is an infrastructure providing services to applications as well as a programming interface to distributed applications. One of these applications includes video games, electronic games played with a controller and return visual feedback to its users. An extension of this is multiplayer games, where many players can play the same game. These players can be human or controlled by the game itself using artificial intelligence (AI). Multiplayer games can be local multiplayer on a local area network (LAN) or online multiplayer on a wide area network (WAN) over a large geographic area. Further extending this is massively multiplayer online games (MMOGs), a large networked game where hundreds or thousands of players connect through the Internet. All of these games have an architecture, or the connection type, working behind the scenes to allow users to connect. However, no networked game will be perfect. Players will be forced to deal with performance issues such as latency and systems must be able to handle the potentially large load of clients connecting to the service. No matter what architecture type the networked game was designed with, there will be performance issues associated with each, both in the logical and physical environments of the network.

II. Importance

Why are the types of networked games important and why should we analyze the performance issues related to online gaming? When designing networked games such as massively multiplayer online games, the overall architecture type of the game must be considered. Failure to consider the type of architecture used when designing can result in issues with scalability, consistency, security, and fast response time for end users, especially with the production of huge network traffic and processing loads created by it. Knowing the architecture types of networked games can allow video game designers to have a better architecture in developing networked games. With a good understanding of the performance issues, developers can mitigate some of these issues that are in their control when developing.

A. Past

The first steps to online multiplayer gaming began in 1973 with the first example of players competing on separate screens with the game *Empire*, "a strategic turn-based game for up to eight players. This was created for the PLATO (Programmed Logic for Automated Teaching Operation) network system... originally built by the University of Illinois" [1]. It is known as "one of the first steps on the technological road to the Internet, and online multiplayer gaming as we know it today" [1]. With the release of Windows 95 and affordable Ethernet cards, networking was brought to Windows PCs, further expanding the popularity of multiplayer LAN games. Between 1993 and 1996, companies attempted to break into online gaming with cable providers, but none of them took off due to slow Internet capabilities and problems with cable providers at the time. The Sega Dreamcast would be the first revolutionary and popular Internet-ready console, containing an "embedded 56Kbps modem and a copy of the latest PlanetWeb browser, making Internet based gaming a core part of its setup..." [1]. However, it was ahead of its time since the Internet was expensive at the time and Sega "footed huge bills as users used its PlanetWeb browser around the world" [1]. It did, however, pave the way for the next generation of consoles, such as the Xbox since "...new console manufacturers learned from and improved the net-centric focus of the Dreamcast, making online functionality an integral part of the gaming industry" [1].

B. Present and Future

Presently, the costs of technology, servers, and the Internet have dropped to the point that fibre-optic speeds are now accessible and common. Internet capabilities have exploded since the early 2000s and "nowadays, many games have an online component that vastly improves the gameplay experience and interactivity" [1]. Recently, games have moved to mobile devices with new challenges to consider - processor speeds, limited memory, and small screens. Although "mobile gaming is already witnessing its first slump" [1], new technologies are actively being developed. The virtual reality and artificial intelligence industries can play a key role in the future of gaming. It provides an opportunity for "fully interactive, 'dynamic worlds' for MMORPG, in which players could move around, interact with other players, and experience the digital landscapes..." [1]. By 2025, gaming could be almost unrecognizable from what it was in the past. "The history of online gaming follows a similar path to that of the Internet... What is going to play a role in the success of gaming is how we deal with network latency, bandwidth, and throughput. Particularly over our mobile networks" [2].

III. Technological Details

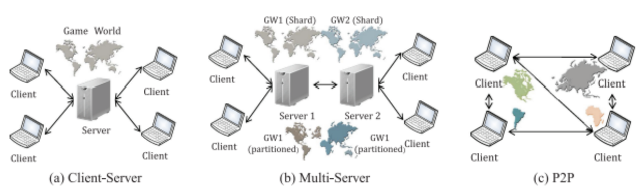

Figure 1. Architecture Types; Source: Adapted from [3]

Networked games can be divided into the architecture types they use. Most networked games use a client-server architecture (with one or more servers), a peer-to-peer architecture, or a hybrid architecture containing features from both. No architecture is perfect, but each has its perks and drawbacks to consider when designing networked video games and as an end-user of the final product. Figure 1 displays the types that will be discussed in detail.

A. Client-Server Architectures

The client-server architecture is the prevalent game architecture, where the server handles "game execution and game state dissemination" [3]. In Figure 1, the client-server architecture is illustrated by (a), where clients connect to a single server. The server holds master copies of all changeable objects and maintains global knowledge of the game world. These master copies are distributed to the clients as replicas. From a design perspective, servers are easy to manage, simple, and controllable. Clients connect to this server to receive information about the game world. In general, servers distribute the load to each connected client. However, this architecture type struggles when the server is hosted in a poor location, causing gameplay to "become laggy, choppy, and poor quality; most home connections can only handle 4-5 remote players at max" [4]. In addition, servers are not scalable since they can only handle so many people and are a single point of failure. Backup servers help solve the single point of failure issue, but maintaining these adds more costs and further decreases the scalability of the entire system.

Multiple servers, or a Multi-Server architecture, can be added as seen in Figure 1, illustrated by (b). Here, multiple servers can handle pieces, or shards, of the game world or take a partitioned approach, where each server is regional and holds an instance of the game world. In the shard approach, it is useful for interest management (players have limited vision, or Area of Interest [AOI], so they interact with objects in their proximity) and load balancing, but needs a hand-off mechanism between servers that is transparent to the players. In the partitioned approach, players are locked to their regional instance of the game world. In general, multiple servers are more scalable and allow high fault tolerance since there is no single point of failure. However, the hand-off mechanism is complicated for consistency, the game world cannot be continuously divided into shards to handle the load of players, and a single region can have issues that a singular server would face. Maintaining these server farms is extremely expensive since "acquiring servers for 30,000 simultaneous players can be about $800,000 and the bandwidth costs can reach hundreds of thousands of dollars. Maintenance costs such as cooling can match these numbers" [3].

B. Peer-to-Peer Architectures

An additional architecture type is the peer-to-peer architecture, illustrated by (c) in Figure 1. Peer-to-peer architectures are distributed and have a collaborative nature. All the connected peers act as both a host and a client, where one could be responsible for maintaining master copies of game objects and updating the replicas. This is distributed among all the peers and with every new peer joining the online architecture, more resources are added to the system. Peer-to-peer architectures can be structured with a distributed hash table (DHT) "as the underlying mechanism for game state distribution and update dissemination" [3] or unstructured where there is no DHT, but "use a mechanism to detect other players in their AOI and directly send updates to these players" [3].

Peer-to-peer technology can coordinate multiple servers, such as "maintaining distributed game state execution and management of server farms or federated servers." [3]. Low latency can be achieved, it has the highest potential for scalability, updates are sent only to interested clients, and if a peer fails, it does not affect many players. "If client nodes communicate directly with each other or perform part of the game state computation, server requirements in terms of computational power and network bandwidth can be significantly reduced" [3]. However, games designed with this architecture type are susceptible to cheating and are harder to manage because there is no centralized server to maintain game knowledge. Updates may be inconsistent due to execution at different sites. Overall, "coordination overhead and complexity is likely to rise" [3] with the number of peers in the system.

C. Comparing Architecture Types

Table 1 displays the summary of the advantages and disadvantages of each main architecture type from Figure 1. Depending on the networked game being designed and considerations on how much network traffic it may have, one architecture may be better than the other.

Table I
ARCHITECTURE TYPE COMPARISONS

	Pros	*Cons*
Client-Server	Simplicity Easy to manage Consistency control	Scalability Fault tolerance Cost
Multi-Server	Scalability Fault tolerance	Isolation of players Complexity Cost
Peer-to-Peer	Scalability Cost Fault Tolerance	Harder to develop Consistency control Cheating

*Source: Adapted from [3]

D. Hybrid Architectures

Server-based architectures can be combined with a P2P architecture. Several approaches to networked games of this type include cooperative message dissemination, state distribution, and basic server control. *Cooperative message dissemination* has the game state "maintained by one or multiple servers but update dissemination uses a P2P approach" [3]. The server then uses a "P2P multicasting mechanism" [3] to send updates to the player, thereby reducing bandwidth and costs on the server side. *State distribution* has the game state distributed among peers, holding the primary copies of objects and responsible for the execution of player actions. "However, all or part of the communication between peers may be managed by the servers" [3] and centralized operations such as authentication are handled by the servers. By distributing the cost of state execution among clients, scalability can be achieved. In summary from above, *basic server control* involves the servers performing centralized operations such as coordinating peer interactions and keeping tracking of sensitive information such as logins and authentication.

E. Performance Issues

In all architectures for networked games, there are unavoidable performance issues due to the varying client hardware and networks. As stated by [2], primary concerns in the future of gaming and its performance will be how network latency, bandwidth, and throughput are dealt with, especially in mobile networks.

Latency: Latency is known as delay in a networking environment, or how long for the data to reach its destination. Claypool and Finkel [5] performed research on the effects of latency on player performance in cloud-based games. Their analysis performed studies measuring the effects of latency using a commercial cloud-based game system and an academic cloud-based system. Each system had users play short game sessions blind to the controlled amounts of latency and measure the user's in-game performance. They found that when network latency is higher, player performance degrades. "Applying a linear regression to the data finds that every 100 milliseconds of latency results in about a 25% decrease in player performance" [5]. Figure 2 summarizes their results

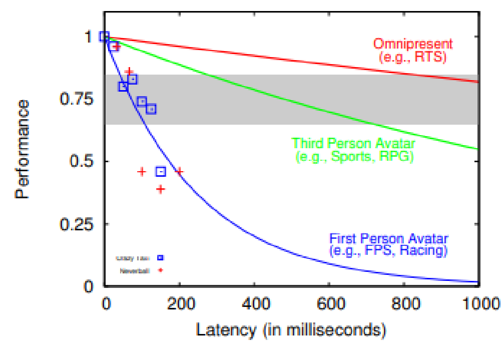

Figure 2. User Performance vs. Latency for Game Genres; Source: Adapted from [5]

when *Crazy Taxi* and *Neverball* were played. In the figure, "the horizontal gray rectangle is a visual indicator of user tolerance for latency" [5]. Generally, gameplay quality is acceptable above this area and unacceptable below it. Depending on the game and the opinion of the user, the exact latency threshold will vary. In addition, "...games with a first-person perspective are more sensitive to latency than games with a third-person perspective" [5]. They conclude that cloud-based game providers must be aware of latency for all games and more research may be needed to study and develop new techniques to compensate for latency in cloud systems.

Lag: Additionally, high latency can lead to lag, or the delay in exchange of game state data, resulting in players' inability to observe and control game actions and making it difficult to perform well and remain immersed with the game. Howard et al. [6] found that "one player's lag can have a cascading impact on the Quality of Experience (QoE) of others... Instead, the primary objective should be to reduce the lag of the most lagged player within each cooperative group" [6]. Lag can be mitigated using common lag mitigation techniques: "behaviour prediction, time delay, time warp, and visual trickery" [6]. Behaviour prediction adjusts the game state to predicted states before receiving the state update, making the game appear continuous despite delayed state updates. Time delay "introduces artificial delay to equalize the latency experienced by all players" [6] and time warp "reorders user input based on client timestamp, rather than arrival time on the server" [6] - both improve fairness but lower immersion and engagement. Visual trickery is a "term for techniques that manipulate the game world to mask delay of game state updates" [6], such as visualizing enemy hits before deducting enemy health on the server. Visual trickery helps with low levels of lag but may result in state misrepresentation with higher levels of lag. In design, lag can be reduced through architectural factors (ex. being closer to the server) or matchmaking players "with low relative latency" [6] together.

Bandwidth and Throughput: Bandwidth is the maximum amount of data a network can handle at a given time while throughput is the amount of data a network can successfully process at a given time. A factor in the difficulty of providing a consistent view of the game world to all players is limited bandwidth, "meaning there is sometimes more data to be updated than can be transmitted without delays" [7]. Techniques can be used to maintain consistency, as mentioned in the *Latency* section. These techniques, however, can have side effects when correcting the inconsistency, such as jarring effects or prolonging the inconsistency. Savery and Graham [7] studied how noticeable consistency corrections are to end-users and if they found the error annoying. Their study found that "inconsistencies in video games are a real problem, and that corrections can be both noticeable and annoying for game players... corrections were more problematic when they interfered with the player's progress in the game" [7]. However, it will always take time to transmit state updates over a network, meaning the game is never perfectly synchronized.

Networks: Advancements in network technology and cost-efficient hardware allowed the placement of servers throughout the world, residing at "the edge of cloud network infrastructures and vastly improved network quality" [8]. To reduce the burden on network performance from the reliance on a single server used by traditional networks, edge networks push the computation to servers closer to end hosts. However, "individual edge servers suffer from capacity constraints during peak traffic hours and encounter problematic network and hardware performance issues" [8]. If MMOGs are pushed to an edge server, they will strain resources due to high traffic demand and additional hardware not available at the location. In addition, mobile platforms need "special mechanisms to handle higher faults and increased latency tolerance, while achieving consistency without additional computational costs" [8]. With more mobile games in development, the demand for closer servers for online network requirements increases.

IV. Discussion

In discussing the architecture types for networked games and performance issues related to online gaming, it is true no architecture will be perfect - each type has perks and drawbacks when compared to one another. In addition, performance issues are concerns to think about when designing networked games. As stated by [2], latency, bandwidth, throughput, and our networks are things to consider for the future of online gaming. In describing the types of networked games, the cost is important when considering the architecture. Most projects have a budget to them, making cost a key concern when designing networked games. Client-servers are simpler to work with but expensive while peer-to-peer architectures are cheaper but difficult to manage. Depending on the budget of the project, one architecture may be preferable over the other - this is up to the designer to decide which architecture type will work best for their networked game. When analyzing the performance issues, the quality of experience for players is important. Issues in connecting and lagging game states can result in lower player performance and satisfaction. These issues can be mitigated with a variety of techniques. Still, the implementation of these techniques should be done carefully to ensure no other bugs are introduced and no other concerns negatively impacting immersion and experience are introduced while implementing.

I believe that performance issues will always exist and that there will not be an architecture that solves all the issues of existing types of networked games. Having experienced designing a networked game with a group for a prior course, I understand performance issues and architecture designs that should be considered when designing a networked game. Concerns I had dealt with were the connection of clients, consistency, and game state of objects - leading to long periods of debugging to ensure it was high quality work and enjoyable for the end-user. There is not going to be one single architecture that will solve all the issues from the client-server and peer-to-peer architectures and performance issues all at once, since this would involve a huge amount of varying factors, such as the type of hardware and quality of network connection for end-users. I feel that taking a peer-to-peer approach for games would be best - it provides a scalable solution, is less costly, and can tolerate connection loss. It may take more time to develop, ensure consistency control, and prevent cheating, but all of these can be mitigated in designing the networked game. With performance issues, I agree with [6] that the goal should be to mitigate the lag for the most lagged player. If one player is heavily lagged, it not only affects that player, but all other players trying to interact with the lagged player, resulting in misrepresentation of state when interacting with them and loss of engagement. By mitigating the lag, all players can interact with each other with high consistency in state and still remain engaged.

Conclusion

Thus, networked games have architecture types that can be used and performance issues to consider when designing. The primary architecture types are the client-server and peer-to-peer architectures, although they can be modified to fit the networked game's needs. Client-server architecture involves clients connecting to a server to receive game data - it is simple to manage but is expensive, a single point of failure, and not scalable. In contrast, peer-to-peer architectures involve peers connecting to each other without a server - peers can distribute game data, making it scalable, cheaper than client-server, and tolerate a loss of connection. However, it is susceptible to cheating on the user's end and is complicated to manage due to no centralized server managing consistency. In all architectures, performance issues will be inevitable. Some key issues for the future of online gaming that should be analyzed include latency, bandwidth, and throughput, especially for mobile networks. No architecture, mitigation, or solution will be perfect, so the implementation will depend on the situation and the scope of the networked game being designed.

REFERENCES

[1] R. Chikhani. "The History of Gaming: An Online Community" *TechCrunch*. https://techcrunch.com/2015/10/31/the-history-of-gaming-an-evolving-community/ (accessed June 20, 2022).

[2] Datapath.io. "The History of Online Gaming" *Medium*. https://medium.com/@datapath_io/the-history-of-online-gaming-2e70d51ab437 (accessed June 20, 2022).

[3] A. Yahyavi and B. Kemme, "Peer-to-Peer Architectures for Massively Multiplayer Online Games: A Survey", *McGill University*, vol. 46, no. 1, pp. 1-51, Oct. 2013, doi: 10.1145/2522968.2522977.

[4] M. Jones. "Network architecture Types in Online Games and How Do They Affect You?" *GamingWeekender*. https://www.gamingweekender.com/network-architecture-types-online-games-affect/ (accessed June 19, 2022).

[5] M. Claypool and D. Finkel, "The Effects of Latency on Player Performance in Cloud-Based games," 2014 13th Annual Workshop on Network and Systems Support for Games, 2014, pp. 1-6, doi: 10.1109/NetGames.2014.7008964.

[6] E. Howard, C. Cooper, M. P. Wittie, S. Swinford and Q. Yang, "Cascading Impact of Lag on Quality of Experience in Cooperative Multiplayer Games," 2014 13th Annual Workshop on Network and Systems Support for Games, 2014, pp. 1-6, doi: 10.1109/NetGames.2014.7008965.

[7] C. Savery and N. Graham, "Reducing the Negative Effects of Inconsistencies in Networked Games," In Proceedings of the First ACM SIGCHI Annual Symposium on Computer-Human Interaction in Play (CHI PLAY ’14). Association for Computing Machinery, New York, NY, USA, 2014, pp. 237–246. https://doi.org/10.1145/2658537.2658539

[8] J. Plumb, S. Kasera, 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 (FDG ’18). Association for Computing Machinery, New York, NY, USA, 2018, Article 2, pp. 1–10. https://doi.org/10.1145/3235765.3235785