Journal of Computer Science and Technology Studies
ISSN: 2709-104X
DOI: 10.32996/jcsts
Journal Homepage: www.al-kindipublisher.com/index.php/jcsts

JCSTS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

Comparison of Open-Source Networking Libraries for Unity Engine in Higher Education

Raymond Leonardo Chandra[1] ✉ Patrick Querl[2], Djamel Berkaoui[3], Koen Castermans[4] and Univ.-Prof. Dr.-Ing. Heribert Nacken[5]

[1234]*Scientific Research, Academic and Research Department Engineering Hydrology, RWTH Aachen University, Aachen, Germany*
[2]*University Professor, Academic and Research Department Engineering Hydrology, RWTH Aachen University, Aachen, Germany*
Corresponding Author: Raymond Leonardo Chandra, **E-mail**: chandra@lfi.rwth-aachen.de

| ABSTRACT

This study presents a comparative analysis of Open-Source Software networking libraries for the Unity 3D engine in developing Virtual Reality applications for higher education. The performance of three OSS libraries—Mirror, FishNet, and Netcode—against the commercial Photon Fusion was evaluated. Performance measurements such as user capacity, performance and cross-platform capability were used as comparison. While Photon Fusion excels in performance compared to other OSS libraries, alternatives like FishNet present a cost-effective and customizable option, ideal for educational settings with particular needs or constrained budgets. The result of this study assists in selecting suitable networking solutions for VR educational applications, promoting their broader adoption in learning environments.

| KEYWORDS

Virtual Reality, Education, Open-SourceSoftware, Computer Networking, Multiplayer.

| ARTICLE INFORMATION

ACCEPTED: 15 July 2024 **PUBLISHED:** 07 August 2024 **DOI:** 10.32996/jcsts.2024.6.3.7

1. Introduction

Virtual Reality (VR) aspects have undergone great advancements in the past few years (Kim & Im, 2022). Aside from the growth in the technological development of the head-mounted display (HMD), controllers and other VR-related hardware, VR applications have also seen an increase in terms of application and content being released in many aspects of the field (e.g. movies, games, teaching and learning, etc.) (Alfarsi et al., 2021). As a result, VR has become much more affordable and accessible to consumers and researchers (Xie et al., 2021). Higher education institutions are also attempting to integrate VR into their teaching and learning practices (Nooriafshar et al., 2004). The VR teaching method has demonstrated its effectiveness for certain teaching activities due to an increase in student engagement and motivation (Nooriafshar et al., 2004). In addition, the engagement between the lecturer and students has been identified in multiple articles as an important teaching and learning factor during the learning experience, either in physical space or virtually (Birt & Vasilevski, 2021; Savickaite et al., 2022; Drey et al., 2022.

The incorporation of multiuser functionality within Virtual Reality (VR) environments has shown potential to enhance motivation and engagement during learning activities (Sykownik et al., 2023). The multiuser enables collaborative experiences in VR which creates an engaging learning environment for students. Moreover, the multiuser aspect of VR helps develop social skills through simulated interactions and teamwork scenarios (Wienrich et al., 2018). Additionally, multiuser VR applications enable distance learning, which can remove the geographical limitation and allow learners from diverse locations to participate collaboratively (Meer et al., 2023). Thus, the integration of multiuser features represents a significant addition to learning within VR environments.

Despite the advantages that multiuser VR has to offer, unfortunately most of the VR applications that are targeted for education are usually single user applications which seem to prioritize personal experience over social experiences (Stephanidis et al., 2020;

Stecula, 2022). For example, a VR application is mostly used solely for observation via the use of 360° pictures and videos with very limited interaction. In another case, the student will wear a VR headset and interact with the VR world by himself while the lecturer will observe and give guidance to the student via screen share of the VR view in the real world (Stephanidis et al., 2020). Even though modern VR headsets nowadays have enough power to run multiuser VR applications close to real time, improper network configuration & unoptimized networking code can cause issues like slow data transfer rates which can impair the user's immersion and cause delays in system responsiveness (Vlahovic et al., 2022). Furthermore, if the university decides to use a third-party service to handle networking for their VR application, they may face data privacy conflicts with the university policy and moreover, they may require additional financial resources for continued use of the service. Due to the aforementioned challenges, universities are hesitant to engage in VR multiuser application development (Morgan, 2009; Lean et al., 2006; Radianti et al., 2020).

To simplify the development of multiuser VR applications, a free 3D engine like Unity combined with an Open-Source software (OSS) networking library can be used. Unity is a popular 3D engine that has a significant market share of 45% in the field of game development including the VR category (Foxmann, 2019; Ciekanowska, 2021). The use of OSS is of particular significance for teaching and learning in higher education because it allows universities to freely customize the software to meet their objectives at a very low-cost factor and without concerns over third-party data privacy conflicts with the University data privacy policies (Vries et al., 2005; Bi et al., 2021).

In this paper, we study and compare the performance of commonly used OSS networking libraries with their commercial alternatives. Namely, we compare the OSS libraries Mirror, FishNet and Netcode with the broadly used closed source software Photon Fusion. We compare each of the mentioned networking libraries in terms of the concurrent user (CCU), FPS (frames per second), CPU usage, bandwidth, round trip time (RTT), and cross platform capabilities factors.

2. Literature Review

The multiuser application boosts user engagement and self-efficacy. Despite the inherent challenges in multiuser development, the process can be simplified through the utilization of networking libraries (Stephanidis et al., 2020). Engelbrecht, (2022) for example, provides a comprehensive guide to developing multiuser games using Unity and Mirror Networking. He explains about the networking fundamentals and the use of Remote Procedure Calls (RPC) and command architecture within Mirror Remote Actions. A multiuser VR walking application which enables real-time, collaborative walkthroughs in a virtual building which enhances the visualization of a building architecture developed by Du et al., (2016), also employs the use of Photon networking and Unity. He et al., (2018) explored significant challenges facing AR and VR network infrastructures. They recommend that these networks should focus on enhancing bandwidth, increasing responsiveness, and reducing computational demands to effectively support AR/VR service. Additionally, Miller & Mansingh, (2015) find an alternative to the traditional client-server model by employing the use of mobile intelligent agents. This approach involves agents migrating to database servers for local query execution and processing, which has shown promising results of improved responsiveness and speed over traditional methods. None of the mentioned articles provides a comparative analysis of OSS networking libraries in Unity.

3. Methodology

This research employed a dedicated server architecture, commonly found in multiuser applications. The server operates as a headless server which maximizes resource utilization by eliminating the need for a graphical user interface (GUI) and any rendering of 3D objects. This allows the server to focus more efficiently on essential tasks such as communication and synchronization.

The efficiency of three prominent OSS networking libraries for Unity which are Mirror, FishNet, and Netcode and a commercial, closed-source library Photon Fusion were evaluated and compared in terms of CCU, FPS, CPU usage, bandwidth, RTT, and cross-platform capabilities. Through this comprehensive analysis, our goal is to offer educators and developers valuable insights that can aid them in selecting the most suitable networking solution for their VR educational applications.

3.1 Client-Server Architecture

The server architecture used in this project is a widely used concept for multiuser applications called a dedicated server (Engelbrecht, 2022). A dedicated server for multiuser applications is a specialized server used for hosting multiuser game sessions. Unlike peer-to-peer (P2P) networking, where clients can host and participate in multiuser sessions simultaneously, a dedicated server is a standalone system responsible for managing and verifying the game's synchronized data and interactions. This architecture plays a crucial role in ensuring a stable and fair usage environment, especially in applications with a large number of users or, for example, in competitive gaming scenarios. The dedicated server tick rate is set to 30Hz. The tick rate is the number of times per second that the server refreshes and processes data. This indicates that it can send packets to clients up to 30 times per second, which include updates on the game state such as player's variables like player position and rotation. As describe in Figure 1, the architecture of the dedicated server is illustrated in a simplified way. The client, who is a user, sends an action request to the server. The server then assesses the legitimacy of the request and, if confirmed as valid, broadcasts feedback that corresponds to the requested client's action to all connected clients.

Figure1: The Dedicated Server Architecture.

Additionally, the multiuser server application is configured as a headless server. A headless server refers to a server that is operated without a GUI (Engelbrecht, 2022) and rendering 3D objects. Very often it is operated and managed remotely, via command-line interfaces. The absence of GUI in a server reduces the consumption of system resources. Thus, allows more computing power to be dedicated to the server's primary tasks which is the communication and synchronization between the server and clients.

3.2 Hardware

The dedicated server hardware used for this research is Windows Server 2019 with 8 CPU cores (only 1 core was used in this experiment), 16 Gb RAM, and 300 Gb of SSD storage. To assess the performance of the networking code in VR, we decided to use the Meta Quest 2 VR headset. The Meta Quest 2 operates on the Android 10 OS and utilizes the Snapdragon XR 2 as its CPU (Central Processing Unit), accompanied by a GPU (Graphics Processing Unit) of Adreno 650 and 6GB of RAM. It functions as a standalone VR system, meaning that it can operate without the need of a tethered device. All computations and rendering are performed on-device. For further analysis, we monitored the performance of a laptop as a client device. This laptop is equipped with an 8-core i7 processor, an RTX 2070 GPU, a 500 GB SSD, and 16 GB of RAM.

3.3 Parameters Explanation

There are several parameters which are important aspects to be considered when comparing the performance of OSS networking libraries, which are: CCU, FPS, bandwidth, RTT, server tick rate and cross platform capabilities (Engelbrecht, 2022).

CCU (Concurrent Users): is a widely used concept in multiuser applications. This term represents the count of individuals actively using or accessing a service at the same time. Essentially, CCU plays a critical role as a key indicator for any online platform, affecting factors such as server specification, operational costs, user experience, and the overall quality of the service. Proper handling of CCU is, therefore indispensable for maintaining seamless functionality and enabling the expansion of online services.

FPS (Frame per Second): is a term that refers to the number of individual images, known as frames, that are displayed or rendered in one second in a video, game or VR applications. FPS is a key measure of visual fluidity and performance of the applications because it affects the smoothness and responsiveness of the gameplay (Kelly & Kumar, 2022). FPS is often used as a benchmark to measure a software's performance on different hardware setups. It indicates how well the application is running and if the hardware can handle the software's graphical demands. However, for the purpose of this project, the client's FPS has been deliberately limited to a maximum of 60. This cap is imposed to ensure uniform performance across various hardware setups and to facilitate a more controlled and effective monitoring of system performance. Additionally, without such a limitation, there exists a risk of performance spikes. These spikes may occur as the CPU and GPU try to generate a higher FPS, potentially leading to non-uniform performance metrics.

Bandwidth: Bandwidth in the context of multiuser games refers to the amount of data that can be transmitted over a network connection in a given amount of time. The bandwidth is measured in Kilobytes per second (KBps). The measurement in this experiment is taken both from the server and clients (Windows and Quest 2). Bandwidth is an important factor in the performance and quality of the user experience, especially in online multiuser scenarios (Morgan, 2009). Bandwidth measures the volume of data that can be sent from one point to another over a network connection per second. In multiuser applications, constant data exchange occurs between the user's device and the dedicated server or between peers. This includes user actions, session state updates, chat messages, and more. When creating a multiuser application, it is desirable to have a low bandwidth usage, as this reduces the amount of data the CPU needs to process.

Round Trip Time (RTT): RTT refers to the delay or time it takes for data to travel from one point to another in a network (Halbhuber, 2022). Lower RTT indicates a faster connection, which is particularly important in real-time applications like multiuser VR applications.

Figure 2 depicts a simplified version of the concept of bandwidth and RTT. When the client requested an action to move to a certain position, the client sent a message to the server. The size of that message is what determines the amount of bandwidth it uses. In the figure for example the size of the message is 100 bytes. After a certain time has passed, the server will send a response (50 bytes) of either approving or disapproving the client's action. The time that passes between a message being sent by the client and it being received and processed by the target device is called RTT. In this case, the client will then be able to move to the requested position after 100 ms from the initial request with a total of 150 bytes bandwidth used.

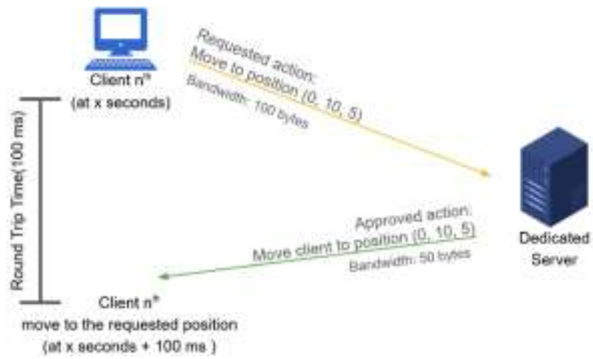

Figure2: A graph of RTT vs bandwidth.

Cross platform capabilities: refers to the ability to create games and applications that can be deployed and run on multiple hardware platforms and operating systems from a single codebase (Patil & Alvares, 2015). For example, the Unity 3D engine is offering extensive support for a wide range of platforms like Windows, Linux, macOS, Android and even WebGL (Web Graphics Library). However not all the networking libraries are able to directly support all platforms that Unity supports. For example, Netcode for GameObjects is not yet able to support WebGL clients.

3.4 Game Engine

The two widely used 3D engines for developing VR applications are Unity [29] and Unreal Engine [30], which use C# and C++ as their programming languages respectively. Both engines are free to download and use in a non-commercial setting. For this research, Unity was chosen because it is one of the most popular 3D engines worldwide, with approximately 45% of game applications, including VR applications, built on it (Foxmann, 2019). Therefore, choosing Unity allows us to reach a wide audience, including many universities. Regardless, the knowledge that is acquired in this paper can be extended to other engines as well. In this paper, Unity version 2022.3.5f1 was used.

3.4.1 Mirror Networking

Mirror was initially released in late 2018 and was designed as a more optimal and efficient successor to Unity's original UNet system (Mirror Networking - Mirror, 2020). Due to its simplicity, effectiveness, and cross-platforms development, Mirror is highly regarded in the indie development community. As a result, Mirror not only receives regular official updates but also benefits from a wide range of community contributions. In this paper, version 73.0.0 of the Mirror was employed for comparative analysis.

3.4.2 FishNet Networking

FishNet is a relatively new networking library in the Unity ecosystem that has quickly gained attention for its balance of simplicity and performance (Fish-Net: Networking Evolved, 2024). FishNet was released for public use in 2021. Designed with the needs of both novice and experienced developers in mind, FishNet provides an easy-to-use interface while still allowing for complex multiuser functionalities. FishNet offers a degree of customization which is capable of handling a large number of players simultaneously. The FishNet networking version 3.4.2 was utilized as a comparison in this paper.

3.4.3 Netcode for Game Objects

Netcode for GameObjects, which was released in 2022, formerly known as MLAPI (multiuser Layer API), is Unity's official solution for multiuser game development (Unity Multiplayer Networking, 2023). Netcode for Game Objects provides a robust, integrated networking system developed specifically for Unity game engine which ensures a seamless multiuser development experience. It is optimized for the server-authoritative model, which is a common requirement in many modern multiuser games. The library is under continuous development by Unity, ensuring it stays updated with the latest features and best practices in network game development. This paper uses Netcode version 1.5.2.

3.4.4 Photon Fusion Networking
Released in 2021, Photon Fusion is a member of the Photon suite, renowned for its robust and scalable multiuser game development tools (Photon Fusion, 2023). Fusion is specifically designed for real-time, high-performance multiuser games. Photon Fusion is noted for its scalability, handling everything from small indie projects to large-scale multiuser environments. Another significant feature of Photon Fusion is its dedicated server support, which is crucial for games requiring stable and continuous online environments. Despite the advantages provided by Photon Fusion, it is important to note that this networking library is not OSS. In this paper, Photon Fusion will primarily serve as a benchmark for comparing other OSS networking libraries. To establish comparison criteria, the study utilizes the Photon Fusion software version 1.1.5 F2 643.

3.5 Multiuser VR Application
he multiuser VR application MyScore, developed by the Academic and Research Department Engineering Hydrology at the RWTH Aachen University, was utilized to assess networking library performance in this study. The choice of MyScore was, primarily due to its OSS nature, allowing for customizable modifications as needed. Additionally, MyScore is built on Unity and was designed under the concept of teaching and learning, targeting higher education institutions. This aligns with the focus of this research paper and made MyScore an ideal candidate for evaluating networking library performance within the context of educational VR applications.

In this study, automated agents were utilized as a substitute for real individuals because assembling around 250 real individuals at the same time posed significant challenges. NPC is a term widely used in games to describe any character that is not controlled by a player (Uludagli & Oguz, 2023) but controlled by the game's system. Notably, an NPC uses the same amount of bandwidth and affects the FPS and RTT similarly to a real user, because the same amount of data is sent over the network. This equivalence ensures that the networking comparison can be relevant and comparable to scenarios involving real human participants.

The NPC’s behavior was designed to simulate real life teaching activities where usually one person speaks to a group and the other just listens (Antonius et al, 2007). Thus, during the test only one NPC is given the task to speak while the rest can only listen. Every NPC however is able to move to any position in the virtual environment. Just like a real VR user whose head and hand movements are tracked and shown in the virtual environment through VR glasses and controllers, NPCs are also able to move their heads and hands. These movements are synced across all users which make teaching simulation in VR more accurate. While the dedicated networking server does not process any graphical information, but only purely textual information, the networking communication performance of a standalone VR headset and PC-based application will be impacted by complicated graphics that the system needs to render (Borromeo, 2024), which includes the virtual environment and NPC’s avatars used in this experiment. Therefore, to accurately assess the performance of the networking libraries, the virtual environment and the NPC’s avatars were designed with simplicity in mind in order to save as much computing power that was intended for the networking comparison as possible. The virtual environment used in this comparison study is a standard plane 3D model and the NPC’s avatars consist of 6 icospheres which represent the head, body, 2 hands and 2 legs of a human individual. The virtual environment and the NPC’s avatars only consist of 2 triangulated meshes for a plane and 20 triangulated meshes for each icosphere respectively. Figure 3 shows the virtual scene and the NPC’s avatars in (A) shaded and (B) wireframe mode that were used during the test.

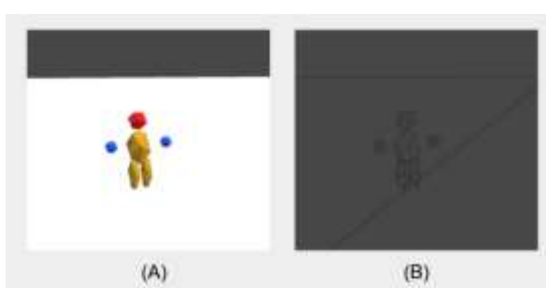

Figure 3: The test scene with one NPC. (A) Shaded mode. (B) Wireframe mode.

4. Results and Discussion

During the networking test comparison, the data was recorded once per second. Additionally, every 30 seconds, an average was calculated from the recorded data. After every 30 second period, an extra of 1 NPC which represents the number of CCU was automatically spawned by the server. Data collection continued until the VR environment contained a total of 250 NPCs, which means that the data recording completed within 2 hours and 5 minutes.

Figure 4 provides a detailed analysis of the bandwidth consumption involved in sending and receiving synchronization data by the server. Notably, Photon Fusion uses the least amount of bandwidth. On the other end of the spectrum, Netcode demands the highest bandwidth, making it the least efficient among the compared networked libraries. Fishnet comes in second place for its low bandwidth usage, followed by Mirror. Moreover, an increase in bandwidth usage is observed as the CCU increases.

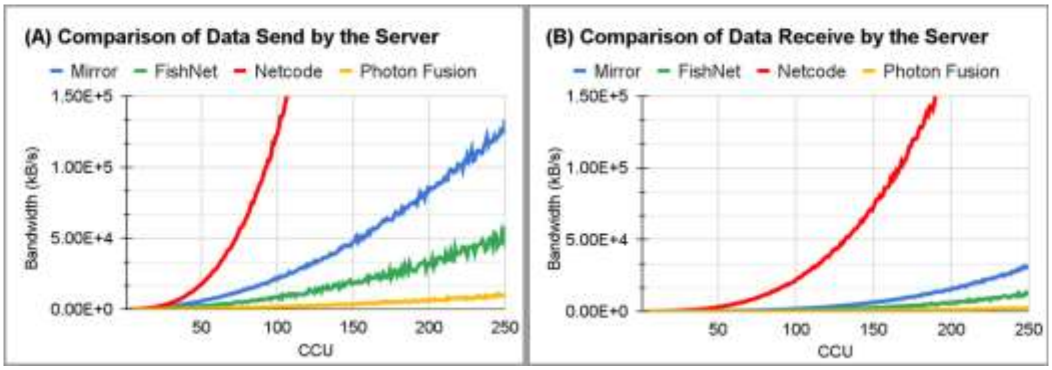

Figure 4: Comparison between the bandwidth used by the server to (A) send and (B) receive the data.

Figure 5 shows how much data is being sent and received by the client. The data sent by a client is relatively constant. Photon Fusion still uses the lowest amount of bandwidth followed by Fishnet, Mirror, and Netcode in that order. On the other hand, Photon Fusion proved at keeping the amount of incoming data low, with Fishnet, Mirror, and Netcode using more data in that order. The synchronization data received by the client is increasing as more CCU is added.

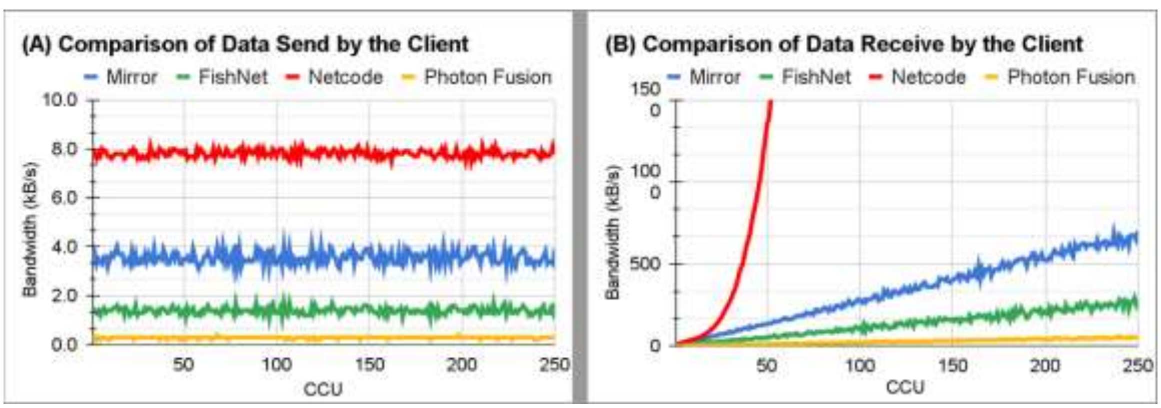

Figure 5: Comparison between the bandwidth used by the server to (A) send and (B) receive the data.

Figure 6 presents the impact of CCU on network RTT across the tested networking libraries. Interestingly, the RTT is quite consistent, with delays in data transmission hovering between 45-55 milliseconds, regardless of the number of CCU connected simultaneously. The consistency may be caused by the networking libraries' efficient network update scheduling and processing. Factors such as update interval, processing time, network infrastructure, and game complexity play a role, rather than just the number of CCUs.

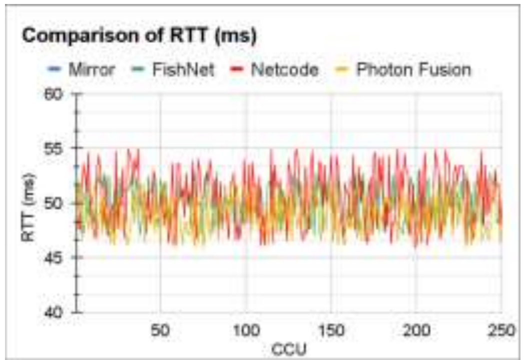

Figure 6: The RTT comparison between the networking libraries.

Figure 7 provides a comparison of CPU usage, in percentage, across different networking libraries on two platforms: Windows and Quest 2. The figure indicates that CPU usage increases with the number of concurrent users (CCU). It is also worth noting that Quest 2 devices failed after roughly 150 CCU due to being unable to handle the computational costs from that point on. Also, the Quest 2 performance is more fluctuating compared to the Windows client. This can cause issues for users as Quest 2 clients try to process more data than the CPU can handle and start to drop data in order to reduce CPU load to keep the application alive.

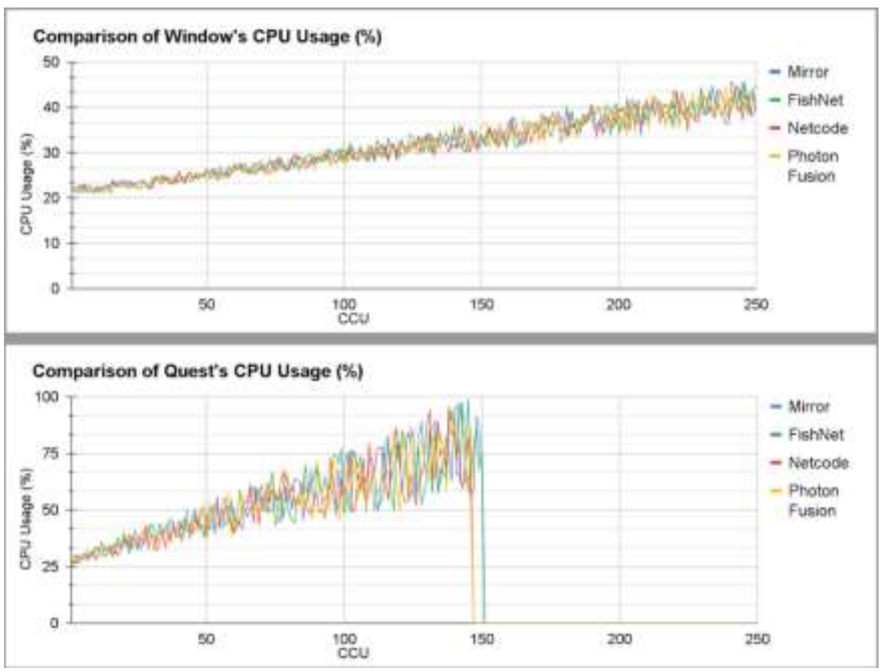

Figure 7: Comparison of CPU load among the networking libraries on Windows and Quest 2 client

The comparison of frames per second (FPS) between each of the networking libraries on Windows and Quest 2 clients is visualized in figure 8. The Quest 2's FPS is seen to be gradually dropping in relation to the CPU usage. As a result, the Quest 2's FPS reaches near 0 when the CCU reaches about 150. The FPS of windows clients start to drop significantly with more than 90-100 CCU existing at the same time.

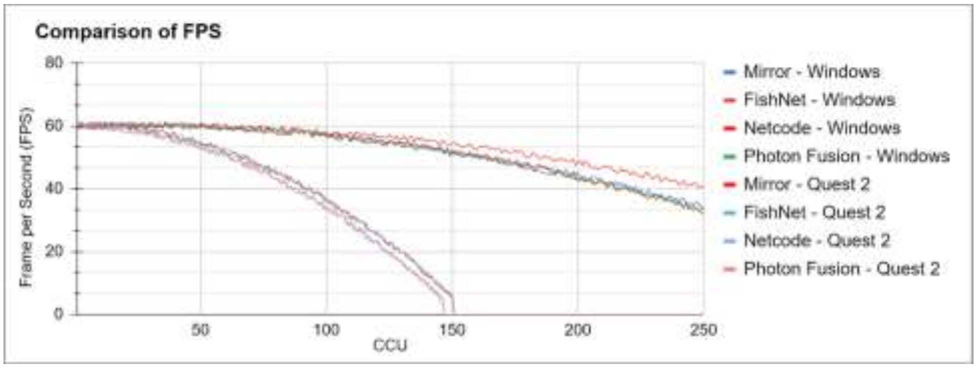

Figure 8: Comparison of FPS between the networking libraries on Windows and Quest 2

Based on the research by Johnson, (2010), Wang & Calvano, (2022) and Kara et al., (2021), classes with more than (>50) students or too few (<24) students usually have lower average test scores. These studies recommend limiting classes at 50 students to potentially boost motivation and focus, which may lead to improved scores. In order to effectively compare the performance of networking libraries in the context of teaching activities in higher education, we have summarized the collected data for 50 concurrent users (CCU), which mirrors the optimal practical classroom size suggested by these researchers. Table 1 presents results of how the networking library performs for 50 CCU and outlines the compatibility of each library across different platforms.

Net working Libraries	CCU	Server (KB/s)		Client (KB/s)		Lat ency (ms)	Windows Client		Quest 2 Client		Client Cross Platform Capabilities
		Send	Receive	Send	Receive		CPU	FPS	CPU	FPS	
Mirror	50	5739.7	240.0	3.5	131.9	46	26	60	42	54	W, M, L, i, A, WGL
FishNet	50	2380.7	101.4	1.4	52.5	52	24	61	48	55	W, M, L, i, A, WGL
NetCode	50	17635.8	2997.5	7.8	1352.1	50	27	61	38	53	W, M, L, i, A
Photon Fusion	50	442.9	20.1	0.3	10.6	46	25	59	40	54	W, M, L, i, A

Note: W = Windows, M = MacOs, L = Linux, i = iOS, A = Android, WGL = WebGL

Table 1: Comparison of the networking libraries for 50 CCU

From the table 1, it can be seen that Mirror sends a significant amount of data (5739.7 KB/s) from the server and receive about 131.9 KB/s for 50 CCU counts, whereas FishNet and Photon Fusion are more conservative in server data sent, with 2380.7 KB/s and receive 101.4 KB/s respectively. NetCode, on the other hand, shows the highest data exchange on both ends—17635.8 KB/s sent from the server and 2997.5 KB/s. In terms of the data sent and received by the client, Photon Fusion still uses the lowest bandwidth followed by FishNet, Mirror and Netcode. The Windows and Quest 2 clients for each of the networking libraries does not differ too much in terms of CPU and FPS. The RTT is pretty similar for all the networking libraries and stays between 46-52 ms. In terms of cross platform compatibility from the client side, all libraries support Windows, MacOS, Linux, iOS and Android; however, Mirror and FishNet extend their support to include WebGL, thus providing a broader range of platform options. Additionally, the comparison result of FPS and CPU usage for Windows and Quest 2 is relatively similar for each of the networking libraries.

The comparison shows that Photon Fusion proved to outperform the OSS networking libraries. However, it is not an OSS and only free up to 20 CCU. Afterwards, there are costs in relation to the number of CCU. This means that in addition to self-hosting the Photon Fusion server, Universities may require to pay an additional amount of money to provide the service.

Arguably by using one of the OSS networking libraries, the cost to run VR multiuser applications can be reduced. Self-hosting the dedicated servers can help mitigate third party conflicts with University data privacy policies. According to the networking comparison, Fishnet performed better than the other two OSS networking libraries. Mirror comes second and is followed by Netcode. Overall Fishnet uses the lowest bandwidth both in server and clients which, in return, allows for more CCU to exchange data before the network infrastructure is overloaded. If one decides to use Netcode, the lack of compatibility with WebGL is to be considered. Conversely, Mirror and FishNet do support WebGL clients as well as Windows, MacOS, Linux, iOS and Android operating systems.

In conclusion, although Photon Fusion has the best performance, it is not an OSS networking library. Between the three compared OSS networking libraries, Fishnet shows a superior performance. Nonetheless, this research acknowledges that there might be some potential of the networking libraries results that we may have overlooked due to an extensive functionality of the libraries. The main purpose of this analysis is to provide developers or institutions with more information before deciding which networking libraries they want to choose. Nonetheless, the experience of those who use the mentioned networking libraries may vary. A reason can be that, depending on the developer's proficiency, they may be able to find more efficient and optimized ways to optimize a certain networking library. Not forgetting to mention, if one decides to pick a library that they are not familiar with, they may need to put some time into learning the libraries as those libraries are built differently.

5. Conclusion

The field of VR has experienced remarkable growth in recent years, significantly impacting various sectors, including entertainment, education, and research (Kim & Im, 2022). In the sector of higher education, institutions are increasingly integrating VR into their pedagogical approaches, with notable benefits in student engagement and motivation. However, a gap exists in the development and utilization of multiuser VR applications for educational purposes (Alfarsi et al., 2021). Most VR applications for education focus on individual experiences rather than collaborative or social ones (Stecula, 2022). Students are mostly restricted to passive observation, such as viewing 360° pictures and videos, or having solitary interactions within a VR environment under the supervision of a lecturer (Stephanidis et al., 2020; Drey et al., 2022). Despite the technological capability of modern VR headsets to support real-time multiuser applications, challenges such as network configuration issues, unoptimized networking code, RTT problems and the potential conflicts with university policies and the financial implications of using third-party networking services further discourage universities from developing multiuser VR applications (Vlahovic et al., 2022; Radianti et al., 2020).

Addressing these challenges, this paper proposes the use of Unity game engine, in conjunction with OSS networking libraries for developing multiuser VR applications. This approach not only aligns with the educational objectives of higher education institutions but also allows for customization and scalability (Foxmann, 2019; Bi et al., 2021). In our study, we analyze and compare the effectiveness of different OSS networking libraries (Mirror, FishNet, and Netcode) that are compatible with Unity, in comparison to the closed-source Photon Fusion. The main objective of this research is to assess various performance measures such as CCU, FPS, CPU usage, bandwidth, RTT, and cross-platform capabilities. This thorough evaluation aims to provide educators and developers with valuable insights to assist them in choosing the most appropriate networking solution for their VR educational applications.

In the comparison, Photon Fusion is found to perform better than the other three networking libraries. However, it is worth noting that Photon is a commercial solution which may raise concerns about costs and data policies for universities. On the other hand, OSS networking library solutions offer a cost-effective alternative with lower bandwidth usage, which could be a critical factor in preventing network overloads in data-intensive environments like VR. Mirror and FishNet both offer extensive platform support, but if a university leans towards using NetCode, it must account for the absence of WebGL support.

The findings in this paper may have limitations and are meant to help developers and institutions make informed decisions about networking libraries. User experience with these libraries, which is not quantifiable in this analysis, along with the developer's familiarity and the learning curve associated with each library, should also influence the final decision in choosing the appropriate networking libraries for their VR multiuser project. Moreover, this study assesses only three most popular OSS libraries in comparison to a single commercial option which potentially missing other viable alternatives. Due to every university has different networking setting, the focus on the specific metrics in this study, neglects important aspects such as security. Furthermore, the use of NPC insteal of a real person in the test environment may not fully capture the real-world situation. For example, factor such as varying internet connectivity and hardware among the students was not reflected in the study.

To address these limitations, future research should include a wider range of networking libraries for comparison which may lead to a more comprehensive analysis. Additionally, incorporating additional evaluation metrics such as security or anti-cheating can highlight the libraries' effectiveness. Conducting tests across different hardware configurations and involving real users instead of NPC will better reflect real-world conditions and challenges.

Funding: The support of the Federal Ministry of Education and Research (Bundesministerium für Bildung und Forschung - BMBF) & German Academic Exchange Service (Deutscher Akademischer Austauschdienst - DAAD) through the funding of the MyScore is gratefully acknowledged.

Conflicts of Interest: The authors declare no conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

References

[1] Alfarsi Al, G. M., Yusof A. M. and Jabbar J., (2021) The practicality of virtual reality applications in education: Limitations and recommendations, unknown, Jul. 31, 2021. https://www.researchgate.net/publication/353907306_The_Practicality_of_Virtual_Reality_Applications_in_Education_Limitations_and_Recommendations

[2] Antonius S., Haines C., Jensen T. H., Niss M., and Burkhardt H., (n.d) Classroom Activities and the Teacher, Modelling and Applications in Mathematics Education, 295–308, doi: https://doi.org/10.1007/978-0-387-29822-1_31.

[3] About Netcode for GameObjects, (2024) Unity Multiplayer Networking. https://docs-multiplayer.unity3d.com/netcode/current/about/ (accessed May 03, 2024).

[4] Borromeo N. A. and Salas J. G. G., (2024) Hands-On Unity Game Development: Unlock the power of Unity 2023 and build your dream game. Packt Publishing Ltd, 2024.

[5] Bi T., Xia X., Lo D., Grundy J., Zimmermann T., and Ford D., (2022) Accessibility in Software Practice: A Practitioner's Perspective, ACM Transactions on Software Engineering and Methodology, 31, 4, 1–26, Oct. 2022, doi: https://doi.org/10.1145/3503508.

[6] Birt J. R. and Vasilevski N., (n.d) Comparison of single and multiuser immersive mobile virtual reality usability in construction education, International Forum of Educational Technology and Society, Mar. 28, 2021.https://www.researchgate.net/publication/350835596_Comparison_of_Single_and_Multiuser_Immersive_Mobile_Virtual_Reality_Usability_in_Construction_Education

[7] Ciekanowska A., Kiszczak – Gliński A., and Dziedzic K., (2021) Comparative analysis of Unity and Unreal Engine efficiency in creating virtual exhibitions of 3D scanned models, *Journal of Computer Sciences Institute,* 20, 247–253, Sep. 2021, doi: https://doi.org/10.35784/jcsi.2698.

[8] Drey T. et al., (2024) Towards collaborative learning in virtual reality: A comparison of co-located symmetric and asymmetric pair-learning, in CHI Conference on Human Factors in Computing Systems, Apr. 2022. Accessed: May 03, 2024. [Online]. Available: http://dx.doi.org/10.1145/3491102.3517641

[9] Du J., Shi Y., Mei C., Quarles J., and Yan W., (2016) Communication by interaction: A multiplayer VR environment for building walkthroughs, in Construction Research Congress 2016, May 2016. Accessed: May 03, 2024. [Online]. Available: http://dx.doi.org/10.1061/9780784479827.227

[10] De-Vries F., Attwell G., Elferink R., and Tödt A., (2005) Open Source for Education in Europe, Research & Practise Conference proceedings Heerlen, The Netherlands, unknown, Jan. 01, 2005. https://www.researchgate.net/publication/254912961_Open_Source_for_Education_in_Europe_Research_Practise_Conference_proceedings_Heerlen_The_Netherlands_November_14_and_15_2005

[11] Engelbrecht D., (2022) Building multiplayer games in unity. Berkeley, CA: Apress, 2022. Accessed: May 03, 2024. [Online]. Available: http://dx.doi.org/10.1007/978-1-4842-7474-3

[12] Foxman M., (2019) United we stand: Platforms, tools and innovation with the Unity game engine, Social Media + Society, vol. 5,4, 205630511988017, Oct. 2019, doi: 10.1177/2056305119880177.

[13] HCI International (2020) – Late Breaking Papers: Cognition, Learning and Games. [Online]. Available: https://link.springer.com/content/pdf/10.1007/978-3-030-60128-7.pdf

[14] Halbhuber D., (2022) To Lag or Not to Lag: Understanding and Compensating Latency in Video Games, Nov. 2022, doi: https://doi.org/10.1145/3505270.3558364.

[15] He D., Westphal C., and Garcia-Luna-Aceves J. J., (2018) Network Support for AR/VR and Immersive Video Application: A Survey, in Proceedings of the 15th International Joint Conference on e-Business and Telecommunications, 2018. Accessed: May 07, 2024: http://dx.doi.org/10.5220/0006941705250535

[16] Introduction, (2024) Fish-Net: Networking Evolved. https://fish-networking.gitbook.io/docs (accessed May 06, 2024).

[17] Johnson I. Y., (2010) Class Size and Student Performance at a Public Research University: A Cross-Classified Model, Research in Higher Education, 51, 8, 701–723, Aug. 2010, doi: 10.1007/s11162-010-9179-y.

[18] Kelly S. and Kumar K., (2022) Unity networking fundamentals. Berkeley, CA: Apress, 2022. Accessed: May 03, 2024. [Online]. Available: http://dx.doi.org/10.1007/978-1-4842-7358-6

[19] Kara E., Tonin M., and Vlassopoulos M., (2021) Class size effects in higher education: Differences across STEM and non-STEM fields, Economics of Education Review, 82, 102104, Jun. 2021, doi: 10.1016/j.econedurev.2021.102104.

[20] Kim D and Im T., (2022) A Systematic Review of Virtual Reality-Based Education Research Using Latent Dirichlet Allocation: Focus on Topic Modeling Technique, Mobile Information Systems, 2022, 1–17, Aug. 2022, doi: https://doi.org/10.1155/2022/1201852.

[21] Lorenzo C.-M., Ángel S M., and Sánchez S., (2012) Studying the effectiveness of multi-user immersive environments for collaborative evaluation tasks, Computers & Education, 59, 4, 1361–1376, Dec. 2012, doi: 10.1016/j.compedu.2012.06.002.

[22] Lean J., Moizer J., Towler M., and Abbey C., (2006) Simulations and games, Active Learning in Higher Education, 7, 3, 227–242, Nov. 2006, doi: 10.1177/1469787406069056.

[23] Morgan G., (2009) Challenges of online game development: A review, Simulation & Gaming, 40, 5, 688–710, Jul. 2009, doi: 10.1177/1046878109340295.

[24] Miller K. and Mansingh G., (2015) Comparing the use of mobile intelligent agents vs client server approach in a distributed mobile health application, *Journal of Computers,* 10. 6, 365–373, Nov. 2015, doi: 10.17706/jcp.10.6.365-373.

[25] Mirror networking, (2024) Mirror. https://mirror-networking.gitbook.io/docs/ (accessed May 03, 2024).

[26] Nooriafshar M., Williams R., and Maraseni T. N., (2004) The use of virtual reality in education, unknown, Sep. 01, 2004. https://www.researchgate.net/publication/228724825_The_use_of_virtual_reality_in_education

[27] Photon Fusion: Setting the Benchmark for Multiplayer games. (2024) https://www.photonengine.com/fusion (accessed May 03, 2024).

[28] Patil P. P. and Alvares R., (2015) Cross-platform Application Development using Unity Game Engine, unknown, Apr. 10, 2015. https://www.researchgate.net/publication/312591645_Cross-platform_Application_Development_using_Unity_Game_Engine Unity real-time development platform, (2024) Unity. http://unity.com (accessed May 03, 2024).

[29] Radianti J., Majchrzak T. A., Fromm J., and Wohlgenannt I., (2020) A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda, Computers & Education, 147, 103778, Apr. 2020, doi: 10.1016/j.compedu.2019.103778.

[30] Sarune S, Millington E., Imants L, Failes J., Kirkwood N., and McDonnell N. (2022)Virtual Reality (VR) Multi-User Lab for Immersive Teaching, PsyArXiv (OSF Preprints), Jan. 2022, doi: https://doi.org/10.31234/osf.io/3w4hv.

[31] Sykownik P., Karaosmanoglu S., Emmerich K, Steinicke F, and Masuch M, (2023) VR almost there: Simulating co-located multiplayer experiences in social virtual reality, in Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, Apr. 2023. Accessed: May 03, 2024. [Online]. Available: http://dx.doi.org/10.1145/3544548.3581230

[32] Stecuła K., (2022) Virtual reality applications market analysis—on the example of steam digital platform, Informatics, 9, 4, 100, Dec. 2022, doi: 10.3390/informatics9040100.

[33] The most powerful real-time 3D creation tool, (2024) Unreal Engine. https://www.unrealengine.com/en-US (accessed May 03, 2024).

[34] Uludağlı M. Ç. and Oğuz, K. (2023) Non-player character decision-making in computer games, Artificial Intelligence Review, Apr. 2023, doi: https://doi.org/10.1007/s10462-023-10491-7.

[35] Vlahovic S., Suznjevic M., and Skorin-Kapov, L. (2022) A survey of challenges and methods for Quality of Experience assessment of interactive VR applications, *Journal on Multimodal User Interfaces*, 16, 3, 257–291, Apr. 2022, doi: 10.1007/s12193-022-00388-0.

[36] Van-der M N., Werf V., Brinkman W.-P., and Specht M., (2023) Virtual reality and collaborative learning: A systematic literature review, Frontiers in Virtual Reality, 4, May 2023, doi: 10.3389/frvir.2023.1159905.

[37] Wienrich C., Schindler K., Dollinger N., Kock S., and Traupe O., (2018) Social presence and cooperation in large-scale multi-user virtual reality - The relevance of social interdependence for location-based environments, in 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Mar. 2018. Accessed: May 03, 2024. [Online]. Available: http://dx.doi.org/10.1109/vr.2018.8446575

[38] Wang L. and Calvano, L. (2022) Class size, student behaviors and educational outcomes, *Organization Management Journal,* 19, 4, 126–142, Feb. 2022, doi: 10.1108/omj-01-2021-1139.

[39] Xie B. et al., (2021) A review on virtual reality skill training applications, Frontiers in Virtual Reality, 2, Apr. 2021, doi: 10.3389/frvir.2021.645153.