Next Generation Networking for Virtual Reality and Game Player/Multiplayer Collaboration

A Fundamental Study to Improve Multiplayer Gaming Experience

Gabriel Taveras
School of Computer and Information Sciences
Florida International University
Miami, FL, USA

Dr. Francisco Ortega
School of Computer and Information Sciences
Florida International University
Miami, FL, USA

Bradley Marzouka
School of Computer and Information Sciences
Florida International University
Miami, FL, USA

Abstract— **The video game industry has seen a tremendous boost in popularity in recent years due to the ambitious number of specifications: graphics, story, action and violent gameplay, online content, etc. What we aim to do is research and implement the future of networking technologies to provide the best quality of service (QoS) within the online gaming environment. By increasing bandwidth, and reducing latency (or lag), using the SDN framework can provide the ultimate gaming experience to gamers across the globe, while having minimal downtime and more throughput. Using SDN can increase the pace of innovation, empower network operations and diversify the distribution of data from client-server interactions. In this paper, we'll discuss about the implementation of SDN within a multiplayer spectrum to provide a better online experience for players, the cyber security aspect for any potential threats, and future executing SDN in future projects that coincide with the future generations of technological advancements within the gaming industry (i.e. Virtual reality gaming).**

Index Terms—**SDN, Multiplayer, QoS**

I. INTRODUCTION

Video games has seen an influx in popularity in recent years, not only among young children, but amongst adults as well; we see a diverse population in the different generations of human beings jumping in to taste the reluctance of an excellent story mode, or just for sheer excitement to compete with other players from around the globe. Concurrently, with more players jumping into the multiplayer world of video gaming, the demand for instant gratification of video game content becomes the decisive action of whether or not players will want to continue progressing within the game. Nowadays, a video game's revenue predominantly comes from the online multiplayer content, and is usually the deciding factor of whether or the not the game will continue forward for future releases, additional patching, and other opportunities to release additional features, adding more of an experience to the game (e.g. sequels, unlockable items, or DLCs). The multiplayer platform is especially remarkable due to the workload the game servers must maintain in order to establish a quality gaming service for the users on a daily basis over extended periods of time. Multiplayer gaming is stretched through a variety of margins to give an adaptable experience for gamers; for instance, depending on the game genre, the workload shall vary, as it requires achieving different functionalities to maintain availability of online for the gamers (e.g. story mode, playthroughs, player collaboration). By using Software Defined Networking (SDN) we have the ability to enable better management and control of communication, data flow, and increase packet distribution within a data center for an optimal experience in quality gaming. Thus, we recommend the implementation of Software-Defined Networking (SDN) to be a foundation layer in next generation gaming to assist in the increase performance of a multiplayer environment. SDN provides an opportunistic methodology to reduce workloads of gaming servers, discover clear and concise network pathways to meet end-user needs and attempting to reduce the failover rate when governing multiple online players on a massive scale.

II. FUNDAMENTALS AND ARCHITECUTAL DESIGN

SDN is the new emerging technological advancement that is becoming the innovative future for running applications and deploying networks. Another name for SDN, commonly referred to as programmable networks, because of its ability to allow applications to run on end systems with dynamic control of a network topology, QoS, improvement of connectivity and routing. SDN has an added benefit of decreasing working costs by usage of uncomplicated software, hardware and management. The potential of applying SDN within multiplayer gaming environment has tremendous capabilities to provide effectiveness for the consumer; the main function for implementing SDN is managing the underlying behavior of the network's runtime. SDN can achieve this by decoupling the control logic a network infrastructure into a centralized controller, which accesses applications via Northbound and Southbound APIs; all in which shall be later clarified in the paper.

The SDN architecture comprises of data and control planes, an integrated controller and a fundamental framework. SDN focuses on several aspects such as disassociation of the control plane and data plane, and the integrated controller with the outline of the network. There are also accessible interfaces among the data plane and devices on the control plane with the capability of the network to be programmed by exterior applications; all of which are also contributing factors that SDN focuses on [1]. With the integration of conventional networking and SDN, we can observe the comparison of the two.

A. Conventional Networking

In traditional networks *Figure 1 part(a)* both the data and control planes are joined onto a network node. The control plan's purpose is to setup the node and compute the routes utilized for data flows. When the routes are computed, they are sent to the data plane [1]. Within the conventional approach, when the forwarding policy has been specified, an alteration can only be made by modifying the configuration of the devices. As a result, this hinders network operators from optimizing performance dependent on several factors such as traffic request, flourishing utilization of mobile devices and the consequences of big data [1].

B. SDN Networking

The results of the limitations on traditional networking grants us insight on a new approach; SDN was developed to meet these demands. As seen the *Figure 1 part (a)*, control is situated in every network node. However, in *part(b)* the control are migrated out of the individual nodes and brought into a single controller. A network operating system or NOS manipulates the SDN switches to gather data using an API. This advises the forwarding plane by supplying a conceptual design of the network infrastructure to the SDN controller accommodating the applications [1]. Subsequently the controller capitalizes on this information to enhance flow management and promote requirements of scalability and adaptability.

III. BENFEFITS OF SDN IN MULTIPLAYER ENVIRONMENT

The architecture of multiplayer gaming expands upon different hardware platforms (PC, laptops, gaming consoles, mobile, etc.), with each fulfilling specific roles upon the capabilities of the deliverable product to the users. Most online games rely on multiple servers, conjoined in a centralized data center; this centralized data environment maintains consistency within a well-established infrastructure to deliver quality gaming in corresponding video frames (ideal graphics is 60 FPS, given no lag), improved redundancy, and

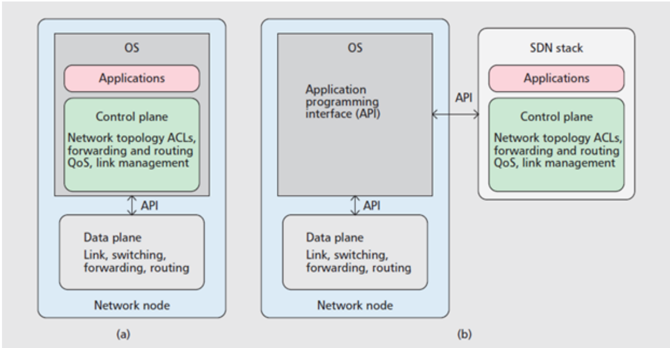

Figure 1: Comparison between traditional and SDN networks: (a) conventional technique. (b) Networking with SDN.

communication of game data between the client-server interaction; all with the end-state to reduce gaming delay, lower latency, and avoid any downtime from an overabundance of packet payloads.

On a global scale, in order to tend the requirement of users across different regions, multiple servers must be deployed to have a better network operation; for instance, US West region can connect to its own dedicated servers, whereas US East will have their own as well, with the same working for European and Asian servers, all with the intent to minimize bandwidth stress compared to a global population attempting to connect to a data center in a single location. Consequently, the more of quality demanded from the customer, the more the company must invest to deliver the best QoS. The mesh of implementing more networking hardware within a data center can prove to be cost effective, as the gaming industry attempts to steer towards an alternative path to provide the ultimate online gaming experience.

Massive multiplayer online (MMO) is a prime example of a game genre where it must be required to handle the payload of having up to millions of online players, actively engaging and conducting real-time content to achieve progression towards a character build of their choosing, or to gain skills points in specific areas to increase their competitiveness. World of Warcraft (WoW) of BLIZZARD Entertainment, Inc. [8] is considered the most popular and recognized MMO game due to its massive in-game content, real-time events, and ability to support and sustain server/ gaming operations for their 7 million subscribers. Other video games (Battlefield, Call of Duty), do not rely heavily in in-game multiplayer content, but must handle the excess load of networking and packet forwarding to decrease latency, and increase the bandwidth to survey a 30 vs 30 conquest match.

A. Static Server

In order to engage in real time multiplayer online games, we must be able to rely on network intercommunication between players within a game; a player's gaming experience is dependent upon specific components, such as: network throughput and reduced network latency. For instance, online games are utilized upon centralized game servers, which operates as a meeting place for gamers. The interactive response of the game is determined by geographical location (regions) and the number of servers in the network. However, video game industries are faced with a dilemma finding ideal locations for the servers due to the quantity increase of players and their locations are undetermined at the time of deployment [2]; resulting in them creating a static game network in an attempt to provide users with satisfactory performance.

B. Custom Tailored Server with SDN

With the increasingly progressive technologies the idea of constructing a custom game network is now conceivable. This game network would have the capability to custom-fit a specific game and its current players. Per other researchers [2], the concept of a custom-built network is proven using SDN in the multi-player environment. The network adapts to a set of specific players. It has been proven that utilizing SDN can enhance a player's response time as opposed to the response time when using a static server.

A software defined network (SDN) can be beneficial over a traditional network in several ways. For instance, if a game server is positioned in an ideal location it would produce a minimum delay which is bidirectional for all game players in a game. The connections between the server and the players may be improved so that when a specific level of play is selected the bandwidth can ensure it. In circumstances of network congestion. It may also be possible to configure the routers between the players and servers so they opt for integral packets (shooting action) instead of non-integral packets (background textures) [2]. The advantage of utilizing a SDN is the game server may be constructed so that the server may be able to migrate to a more optimal position when the participants change. Another benefit is SDN is that it supports sending out multiple unicast flows as opposed to traditional networks that support a single multicast flow [2].

IV. IMPLEMENTAION OF SDN IN MULTIPLAYER ENVIRONMENT

SDN implemented networks have the capability to extensively change the communication process between players and games. In order to configure a particular application for SDN network, a high-level outline is required. This would assist developers when creating application specific SDN network by concealing the tedious technicalities of the SDN network. In the same study [2]; a hypernet can be constructed. A hypernet is defined as a dynamic program that is utilized to create a SDN network for a specific application. The program would incorporate the necessary logic when making selections of resources which the specific application needs. The hypernet would also include the software to be used on numerous nodes in the SDN network including the code for dynamic centralized node with ideal location. To elaborate on the notion, hypernet is based on a virtual application. Similarly, a virtual appliance determines the configuration a virtual machine incorporates exactly what is necessary for the software to execute a specific service. A hypernet is an integrated bundle that covers all the knowledge needed to create a SDN network and deploy the software needed to run on the SDN. For instance, a hypernet tailored for a specific game would contain a program to devise the optimal network topology from a given set of current players, the configuration files needed to use the game server on the server node, the software protocol stacks utilized on the end system, the specific router methods (priority queues, multicast, routing tables) to use and the guidelines needed to setup the configurations and software for the interrelated nodes. Since the hypernet binds the code and knowledge necessary, developers only need to render the program to create an enhanced network for the specific game.

The underlying framework supporting the hypernet is a specific set of APIs which aid in the simplicity of the application. These API calls permit the hypernet o firstly define the topology effortlessly, secondly program the SDN routers and lastly allow players to join the SDN. To further discuss the APIs used we will elaborate one four different calls: *findPR()*, *findCentralNode()*, *loadApp()* and *join()*.

The *fndPR()* command locates a close SDN router based on the connection point of a player. Then the player is connected to the proximate location in the SDN network. The API enables players to deliver their data using the enhanced SDN network to the game server. This API permits players to transmit their data over the improve SDN network instead of having the players use their IP to communicate with the game server. Only the channel between the connection location and the player is transmitted over IP where QoS and the path cannot be controlled. The rest of the path between the game server and the player pass through the SDN network which can be improved for maximum performance.

Th*e findCentralNode()* API call locates the SDN node that is the midpoint to asset of players. This is beneficial for discovering the ideal location to situate the game server. The center node can be selected depending on the number of hops or network efficiency from a specific set of nodes. In the instance of game networks, this is beneficial because a central node produces minimal network delay for all players. For this call to be enforced the Network Hypervisor must gain information on the fundamental SDN provider's network. For example, physical topology information is needed in order to determine a central node with a number of hops. When determining the best central node through network performances, an incisive technique is needed. This is supplied by the SDN provider to detect the distance between two nodes. Ping probing is used to calculate the Round-Trip Time (RTT) including delay between the current players and the most probably centralized node. In order to use time more efficiently ping probing is not performed bidirectionally to all accessible SDN nodes. The available resources in the SDN network is split among a small amount of aggregates. All the nodes in the same collection share the same network region and have relatively the same network performance.

The *loadApp()* loads a configuration and application file into a particular node; an alternative method is to use a command to run the application. This API is utilized to configure and execute the game server on the centralized node. It can also be used to load and run software on other routers in the network. The join API call constructs an IP channel between the corresponding SDN entry node and a player. It appoints an SDN network address to a player and sets up the routing tables accurately to guarantee that the player is capable of communication with other nodes in the network. When a player joins the optimized game network, the client can't talk to the server through the IP allowing the player to start playing.

V. CUSTOM SDN FOR OPENARENA GAMING

To display the hypernet with a custom SDN network, an OpenArena (open source multiplayer online) game is used. This game has numerous available game servers where players connect. Since the game is open sourced, it will permit players to assemble and execute their game server. This provides the advantage of the possibility for a group of participants to determine the ideal location for the game server that would result in the lowest delay for all players and then run their fame server from there. The purpose of the OpenArena hypernet games was to immediately choose the ideal location for the server, execute the game server, configure the SDN network and connect the server to all the players. ProtoGENI was used as the foundation for the SDN provider. This is a prototype implementations of *global environment* which is used for *network innovations* (GENI). It proposes and executes the GENI control framework which includes resource management and allocation for authenticated and authorized experimenters [3]. The network resources in ProtoGENI permits their users to supply network connections between two supervised nodes, which makes it the ideal SDN provider for the SDN game network.

Initially, the hypernet looks at the configuration file for the SDN network. The file contains password necessary to join network, intended players network location, and the name of the SDN network. The hypernet uses the findPR API to locate a close by gateway programmable router for every player. The next step is using findCentralNode API which would create a virtual tunnel from each gateway to central node resulting in minimum delay to all gateways. The loadApp API then loads the OpenArena server app to the central node through the hypernet. Lastly the build API deploys the SDN network on the hypernet.

After these steps the SDN game network is functional. The next step is the use of the join API that the players use to join the game. Players sends their requests to the hypernet and determine the set up for the interrelated gateways (which IP addresses that must be assigned and the entries that need to be appended to the routing table. A generic routing encapsulation (GRE) tunnel is constructed between a player's appointed gateway and a player. A GRE is a tunneling protocol which was developed by Cisco. This protocol envelops numerous network layers protocols within virtual point to point links over an IP network. ProtoGENI permits the users to supply virtual links in-between the controlled nodes. Regarding our

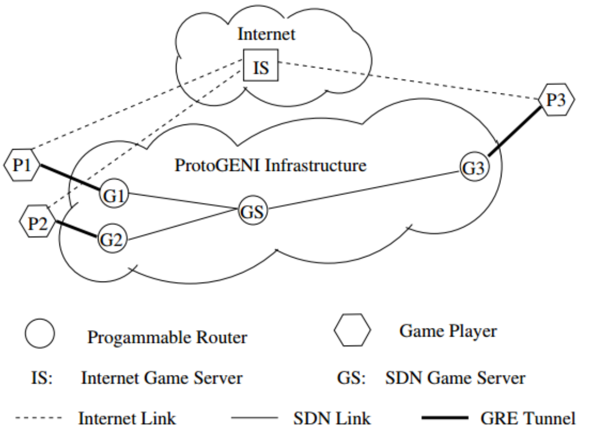

Figure 2: Fundamental setup of ProtoGENI

instance the hypernet retains 100Mpbs bandwidth for the SDN links. This amount is adequate for the peak sending rate for 200Kbps. The hypernet can be extended to permit dynamic game joiners by locating and appointing gateway to new players who are requesting it.

A standard OpenArena is compares to the OpenArena Hypernet game. In the experiment the OpenArena hypernet game was run on the hypervisor and the custom SDN network was devised for 3 players. Figure 2 displays the fundamental setup. Using ProtoGENI the optimal gamer server was found which yielded an RTT of 22ms on average whereas the ideal public Internet Server yielded an RTT of 212ms in average. This proves that utilizing an SDN network can produce a significant enhancement in the game's response time.

VI. FUTURE WORKS WITH VIRTUAL AND AUGMENTED REALITY PLATFORMS

There is a high probability that un-parallel levels of data is transferred across the network and can be temporarily stored locally to guarantee low latency. Virtual reality provides a challenge that is directed on the network. Networks require low latency, additional storage at their data hubs, and copious more centers to allocate content (4). Most global connectivity providers undertake advanced traffic management, backbone capabilities, content delivery networks and cloud footprint. However, connectivity to the end user will probably attest to be the bottleneck as it hinders high quality cloud streaming of VR content. According to the most current State of the Internet Report from Akamai, global connection intermediate connection speeds were as little as 5.6 Mbits/s and 7.1% global connections were over 25Mbit/s [4]. A great deal of work is need to optimize bandwidth speeds to levels needed to maintain VR/AR capabilities.

One technique of reducing latency is migrating content near to the edge of the network. This technique would be tedious as mass cross-country VR broadcast (e.g. Cricket) which has a lot of edges. The concept needed is networking that allows the business and the network operator to have complete visibility; additionally, content and the use of it needs to be controlled despite the housing. A resolution includes utilizing Software-Defined Wide-Area Networking SD-WAN. This would provide an avenue with optimal end user experience. This Software defined architecture is beneficial because it permits network operators and businesses with the capability of manipulating resources and priorities as needed. For instance, if the content at one edge is not being entirely used, with SD-WAN those resources can be allocated to the edge that requires it. It can also be programmed in advance [5].

Furthermore, the WAN can be improved to guarantee that data is being using extraneously meaning that data is not being sent and resent which would result in lessening the volume of traffic on the network. However, this is impractical for a real-time video, data from other application utilizing the same network can be de-duplicated [5]. This means that data is only sent once and then cached. By doing this the available bandwidth can be incremented, latency can be reduce resulting in the impediment of a glitchy feed. Our proposal is to combine application-aware networking, end user experience and end to end visibility of network performance. This will permit concerns to be addressed by detecting and fixing in a more proactive manner. If problems arise with bandwidth for high resolution streaming this can be contested so that operator can reduce the quality minimally while preserving a reliable stream therefore hindering the bouncy motion that results on a nauseated end user. When live VR strikes, prevailing networks would be under an immense burden. Currently Australia has the technological advantage [5], as they are transforming networks based on legacy requirements to a SDN model.

VII. CYBER-SECURITY ASPECT INVOLVING SDN

With current and emerging new technology, comes potential risks that might threaten the network infrastructure of many industries. SDN may solve the latency dilemma with multiplayer gaming, however, its implementation may open a door for new potential malicious threats from cyber criminals. As implementing SDN may have its potential risks, as compared to traditional network capabilities, SDN does offer advantages over its counterparts in perspective of maintain availability and information security. In contrast to regular, conventional networks, SDN does provide the following, in respect to security:

More Granular Security - One downside to the classic style network can be seen in the time it takes IT staff to push security policies out across the network. With SDN, security policies are focused on the SDN controller. This allows enterprises to have access to a single point of control to push new policy out across the network to streamline updates to possible security flaws found on a network.

Central Network Provisioning - As more and more networks are becoming virtualized and more virtual networks become part of physical networks, it becomes more time consuming to manage network dependencies. SDN networks can better provision both virtual and physical networks from a central position. This makes it so that links and dependencies between physical and virtual networks are eliminated.

Better QoS Control - SDN networks are much better designed to handle high bandwidth applications like big data and virtualization. With SDN, administrators now have the capability to better control data flow throughout the network which in turn makes it easier to control quality of service (QoS) for data transmissions. Using an SDN network for collaborative efforts reduces latency and has the capability of guaranteeing data flows better than a more traditional network. This is due to SDN's ability to be able to direct and automate data traffic from the controller to network switches.

Better DDoS Prevention Methods - Using a moving target defense system, an SDN network is better equipped to handle a possible DDoS attack. This is because the network can be configured to have an unpredictable system configuration so that it becomes more difficult for an attacker to successfully attack and take down a network.

A. Potential Malicious Attacks Involving SDN

In this section, we found the potential attacks that could compromise the stability of an SDN network by attacking different layers of the network. In an SDN network, the data plane is kept separated from the control plane. Because of this, one of the biggest weaknesses of an SDN network is its capability of withstanding certain attack.

Application Layer Attack - In an SDN network's application layer, applications communicating share resources made available by the controller. Because of this sharing, an infected application can create resource deficiencies if not caught by other defense mechanisms and can spread to other application in this layer.

Control Layer Attack - An SDN network is mainly controlled by its centralized controller. Because of this, during an attack, this area of the network is primarily targeted. One example of an attack is a flow table overload. This is when a network device encounters a flow on the network that does not match any flow rules on its flow table. This table then, by design, gets sent to the controller to decide where to send. This can be exploited by creating packets from forged IP addresses that can create new flows on the network. Due to an SDN controller's limited memory size, this type of attack can overload the controller and disable access to the network.

API Attack - Unauthorized devices and applications can potentially attack the network from many different angles. It is important to be able to sufficiently protect data flows coming into and leaving the SDN controller. A potential attacker can impersonate a legitimate user and target an entry point into the controller to overload it with data and restrict access to it.

B. Mitigation and Resolution

In this section, we discuss some proposed solutions to the security issues discussed in the previous section. Different solutions help solve a different security issue currently present in an SDN network. Most of the solutions discussed here focus on using protocols.

Hybrid Defense Mechanism - As discussed before, the primary area of interest to attackers is the SDN controller. Because of this, it is important to have a mechanism in place to successfully prevent these attacks from occurring. To protect the controller, a hybrid defense mechanism can be deployed across the network. One possible defense mechanism is the moving target defense mechanism briefly mentioned before. This mechanism dynamically reconfigures a system's resources to make it more difficult for an attacker to accurately pinpoint an attack to the SDN controller. Research needs to be done in this area, however, when it comes to this method as using this approach introduces a potential latency problem into the network by the additional processing time needed to check each data flow traveling across the network.

Additional Controllers to Verify Data Flows - Possibly using multiple SDN controllers linked together can help reduce the potential of memory overflows when verifying a data flow across the network. The main objective with this solution is to create a larger cache to be able to reduce the possibility of a flow table overload attack. Other software based solutions such as FortNOX can be used to reduce the potential number of application layer attacks. Using a specifically designed protocol at each layer of the SDN network can help protect against an intrusion, however, latency may be affected and more research is needed in these areas to verify the impact on latency when communicating across a network

VIII. CONCLUSION

With new emergence of network technologies, it seems just to grant the gaming community a better, more fluid gaming experience. we want to attempt to identify the most common problems that the gaming community mostly experiences when playing online. The purpose for our research to bring about the implementation of SDN as a method to increase the network interaction between the gaming console (PC, XBOX, PlayStation, etc.) to the actual gaming server(s); our overall intent is to grant the gaming world higher data throughput and low-latency online collaboration. But because there are thousands, if not millions, of users attempting to access the same content, the hardware the servers face sees an influx of workload from players across the region, and depending on the size and popularity of the fame, the world. SDN can provide a solution to distribute the workload from within that centralized environment, increase network bandwidth and possible speed interaction, all with the goal to please the gaming community. We want our research to provide the foundational first-step for future implementation for multiplayer collaboration, in the online gaming environment.

Finally, this generation of gaming has seen a tremendous increase in content and in-game material; as hardware capabilities increase in power, it would be just as sensible to invest in better networking technology for the customers. What we would also like to implement in future projects, is to expand SDN onto the virtual and augmented reality platforms (VR, AR). VR headsets (e.g. Oculus, Vive) are the future of gaming and will require more additional, extensive research to cater to gamer community; moreover, the amount of technology, time, and investments required to begin enabling for the AR/VR platform will immense, but will be well worth it for its rapid advancement. For a future university project, we would like to see our colleagues provide a solution and expand upon the networking capabilities involved to give the VR environment a better experience and how users can benefit from SDN's resources. As time and technological innovations pass, it will not belong until we'll be in a world similar to Sword Art Online [9].

ACKNOWLEDGMENT

A special and huge thank you to the student/researchers in the References List. Without their fidelity and perseverance for the interest of SDN, this paper would not be possible. Also, a big thank you to our mentor Dr. Francisco Ortega and our professional colleague, Aarti Ragoonath, for their innovation and insightful assistance with providing proper and thorough SDN formalities and guidance.

REFERENCES LIST

[1] S. Sezer *et al*., "Are we ready for SDN? Implementation challenges for software-defined networks," in *IEEE Communications Magazine*, vol. 51, no. 7, pp. 36-43, July 2013. doi: 10.1109/MCOM.2013.6553676

[2] S. Huang and J. Griffioen, "HyperNet games: Leveraging SDN networks to improve multiplayer online games," *Computer Games: AI, Animation, Mobile, Interactive Multimedia, Educational & Serious Games (CGAMES), 2013 18th International Conference on*, Louisville, KY, 2013, pp. 74-78. doi: 10.1109/CGames.2013.6632609

[3] D. Li, X. Hong and D. Witt, "ProtoGENI, a Prototype GENI Under Security Vulnerabilities: An Experiment-Based Security Study," in *IEEE Systems Journal*, vol. 7, no. 3, pp. 478-488, Sept. 2013.

[4] T. Mastrangelo, "Virtual Reality Check: Are Our Networks Ready for VR?," *Technically Speaking*, 29-Jun-2016. [Online]. Available: http://blog.advaoptical.com/virtual-reality-check-are-our-networks-ready-for-vr. [Accessed: 23-Apr-2017].

[5] I. R. Raper, "Are networks prepared for the live virtual reality experience?," *Computerworld*, 16-Nov-2016. [Online]. Available: http://www.computerworld.com.au/article/610232/networks-prepared-live-virtual-reality-experience/. [Accessed: 23-Apr-2017].

[6] Amiri, Maryam, et al. "SDN-based game-aware network management for cloud gaming." *Proceedings of the 2015 International Workshop on Network and Systems Support for Games*. IEEE Press, 2015.

[7] Gorlatch, Sergei, and Tim Humernbrum. "Enabling high-level QoS metrics for interactive online applications using SDN." *Computing, Networking and Communications (ICNC), 2015 International Conference on*. IEEE, 2015.

[8] “World of WarCraft.” [Online]. Available: http://us.battle.net/wow/

[9] “Sword Art Online” [Online]. Available http://www.swordart-onlineusa.com/

[10] Salazar, Mikel, et al. "Enhancing cybersecurity learning through an augmented reality-based serious game." *Global Engineering Education Conference (EDUCON), 2013 IEEE*. IEEE, 2013.

[11] Minchev, Zlatogor. "Cyber Threats Identification in the Evolving Digital Reality." (2016).

[12] Nunes, B. A. A., Mendonca, M., Nguyen, X. N., Obraczka, K., & Turletti, T. (2014). A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications Surveys & Tutorials*, *16*(3), 1617-1634.

.