IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 18, 2017, accepted September 11, 2017, date of publication September 18, 2017, date of current version October 12, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2752643

SDN-Enabled Game-Aware Routing for Cloud Gaming Datacenter Network

MARYAM AMIRI, (Student Member, IEEE), ASHKAN SOBHANI, HUSSEIN AL OSMAN, (Member, IEEE), AND SHERVIN SHIRMOHAMMADI, (Fellow, IEEE)
University of Ottawa, Ottawa, ON K1N 6N5, Canada

Corresponding author: Maryam Amiri (mamir097@uottawa.ca)

ABSTRACT Cloud gaming or gaming as a service, the newest entry in the online gaming world, leverages the well-known concept of cloud computing to provide real-time gaming services to players. This gaming paradigm provides affordable, flexible, and high performance solutions for end users with constrained computing resources and enables them to play high-end graphic games on low-end thin clients, because it renders everything in the cloud and simply streams the resulting high-quality video to the player. Despite its advantages, cloud gaming's quality of experience suffers from high and unstable end-to-end delay. According to this fact of cloud gaming, datacenters are in charge of performing complex rendering and video encoding computations, delivering high-quality gaming experience to gamers which requires an efficient and smart resource allocation mechanism to allot resources (e.g., memory and network bandwidth) to gaming sessions consistent with their requirements. In this paper, we propose a bi-objective optimization method to find an optimum path for packet transmission within a data center by minimizing delay and maximizing bandwidth utilization. We use a metaheuristic model, called analytic hierarchy process, to solve the NP-complete optimization problem. The resulting method is an analytic hierarchy process-based game aware routing (AGAR) scheme that considers requested game type and requirements in terms of delay and bandwidth to select the best routing path for a game session in a cloud gaming network. The method executes within a software defined network controller, which affords it a global view of the data center with respect to communication delay and available bandwidth. Simulation results indicate that the AGAR can reduce the end-to-end delay by up to 9.5% compared with three other conventional representative methods: the delay-based Dijkstra, the equal cost multi-path routing, and the Hedera routing algorithms. In addition, we demonstrate that the proposed method assigns game flows to network paths and OpenFlow switches in a balanced manner that prevents potential network bottlenecks.

INDEX TERMS Cloud gaming, software defined network (SDN), optimization, analytic hierarchy process (AHP).

I. INTRODUCTION

Recent advances in cloud computing, data center deployment and virtualization technologies bring new opportunities to the gaming industry, and enable end-users to play high-end graphics games on any low-end device [1]. Based on recent industry facts, reported by the Research and Market Institute, revenue from the cloud based video games industry reached 476 million US$ by the end of 2015, and such a market will continue to thrive in the years ahead and is expected to reach 650 million US$ by the end of 2020 [2]. Generally there are two main approaches for online gaming: online gaming based on file streaming and cloud gaming based on video streaming. Although both approaches utilize the client-server architecture, in the former, the server sends updates to the client in response to a change in game context, whereas, in the latter the game server performs game logic processing and rendering, and finally encodes the game graphics into a video that is streamed onto the client. Hence, in the video streaming approach, the game server is responsible for a significant part of the computational operations while the computational load is considerably diminished at the client side. Consequently, with this approach, gamers can make use of thin clients. Therefore, gamers are not concerned with matching gaming devices with the specific hardware requirements of the latest games. Despite the aforementioned advantage, it is reported in [3] that gamers using cloud gaming approximately experience 1.7 times higher latency than the gamers using console games. Therefore, providing cloud gaming services with comparable quality to console games is still challenging.

Tailoring the public cloud infrastructure to the specific requirements of cloud gaming is a relatively challenging task. Also, the results in [4] indicate that public cloud providers such as Amazon are not capable of supporting cloud gaming services with proper QoS provisioning. Hence, cloud gaming service provider companies (e.g. StreamMyGame, Giaki and OTOY) are investing in building their own infrastructure and proprietary platforms to facilitate the allocation of processing and computational resources while meeting the gaming-specific needs.

In our previous work, we proposed a cloud path selection method to reduce end-to-end delay and delay variations (jitter) [6]. The method adaptively disperses the game traffic load among different network paths according to their end-to-end delays. The method uses a global view of the network status as its input and hence employs a Software Defined Network (SDN) controller to direct routing decisions for the streams of gaming data within the cloud. Although this method showed promising results by decreasing the delay experienced by users, it did not consider the specific characteristics of games to select the appropriate routing strategy. Game characteristics vary between different games, and to offer an optimized experience, a solution must consider game specifics. Delay sensitivity often varies based on the game genre. For example, First Person Shooter (FPS) games are more sensitive to delay compared to real-time strategy (RTS) games [7], [45]. So, the tolerable delay depends on game genre. Ideally, routing algorithms must resolve the best routing strategy to satisfy certain Quality of Service (QoS) requirements. In QoS-based routing schemes, routes should be chosen based on features of the transmitted data flows, such as bandwidth requirement and delay sensitivity. There are two main goals that need to be achieved by the QoS-based routing algorithm. The first goal is to find a path that satisfies the QoS requirements [8]. The second goal is to optimize the global network resource utilization. In this work, we propose a bi-objective optimization scheme that strives to jointly minimize the overall delay experienced by users and maximize bandwidth utilization within a data center by considering the requested games genres. The proposed scheme makes use of an Analytic Hierarchy Process (AHP) based Game Aware Routing (AGAR) method to find the initial feasible solution that establishes the optimal path between the core and Top of Rack (ToR) switches in a cloud data center, while taking game-specific characteristics into account.

The rest of this paper is organized as follows: in the next section we discuss the background and related works. In section III we present our proposed optimization framework, namely the AGAR method, followed by a description of our implementation and performance evaluations in section VI. Finally, in section V we conclude the paper and provide directions for future work.

II. BACKGROUND AND RELATED WORKS

In this section, we look at the background and related work in four areas: A) cloud gaming delay, B) relation between network resources and game genres, and C) Data center network (DCN) and software defined networking (SDN).

A. CLOUD GAMING DELAY

Although cloud gaming offers many benefits to both gamers and cloud gaming service providers, it also brings new challenges such as delay and delay variation (so-called jitter). Network delay reduction has been found to have the most significant role in realizing a satisfactory QoE [7], [9], [10].

According to [11], the total cloud gaming end-to-end service delay experienced by users consists of three components: 1) playout, 2) network, and 3) game server delay. Playout delay is the time it takes the client to play the compressed video after receiving it from the cloud. Networks delay consists of data center intra-delay and Internet Service Provider (ISP) delay. Server delay is the time spent by the game server to process the users' commands, render the corresponding video frames, and compress them.

Since the overall delay experienced by users stems from different sources, the proposed techniques to reduce delay can be divided into two main categories. The first category includes methods that improve the game engine and video codec components through techniques of image warping for motion estimation, view compensation, and object information extraction to facilitate video compression [12]. The second category includes mechanisms for network resource provisioning, Virtual Machine (VM) placement, and server selection. Although there is a large body of work that covers the first category of methods, the schemes pertaining to the second category are closer to our work. Therefore, in this section, we will further explore the latter methods.

Several works proposed to mitigate the delay through VM migration and/or dynamic game server provisioning. Jalaparti *et al.* proposed a framework in which the end-to-end delay of each game session is being constantly monitored. Once it reaches a critical threshold, the corresponding game server is moved to a new optimum location to lower the end-to-end delay [13]. Similarly, [14] presented a VM placement solution that aims to maximize the net profit for the service provider while providing users with satisfactory QoE. In this regard, a server selection method was proposed by [15] which aims to minimize the end-to-end latency by assigning optimal servers to a group of gamers within the same geographic area.

In regards to the efficient utilization of DCNs, the results of the experiments conducted by [16] on a DCN comprising 150 inter-switches and 1500 servers, show that despite of existence of many network links, 15% of congestions remained for more than 100 seconds. This rather confirms the in our previous work, we advocated the separation of the control and data planes in the DCN through the use of Software Defined Networking (SDN) techniques to optimize the forwarding strategy within the DCN for delay reduction [6]. Furthermore, in [17], we proposed a method to optimally assign game servers to gaming sessions and select the best communication path within a datacenter considering the type of requested game, current game server loads, and current path delays.

B. RELATION BETWEEN NETWORK RESOURCES AND GAME GENRES

Online games in general have been remarked as delay sensitive applications; however, the maximum tolerable delay varies with the game genre. As described in [9], for First Person Shooter (FPS) games, the users will notice a delay of more than 80 milliseconds while for Real-Time Strategy (RTS) games, the delay of more than 150 ms has a detrimental effect on users' QoE.

It is well established that game design significantly affects network traffic characteristics [18]–[20]. Although the fundamentals of game design are similar for most video games, there are some design aspects that differ from one genre to another (e.g. camera perspective). Suznjevic *et al.* [21] proposed a generic evaluation framework that identifies the influence factors that are relevant for gaming and have an impact on QoE. They identified the following three layers: 1) QoS influence factors; 2) User and system interaction performance aspects; 3) QoE features related to the end user quality perception and judgement processes. Claypool and Mark [22] showed that game design can dictate the required computational and network resources and affect the users' perceived quality score. Moreover, he introduced two indices that are directly relevant to QoS parameters: the average of Intra-coded Block Size (IBS) and percentage of Forward/backward or Intra-coded Macroblocks (PFIM) to measure the scene complexity and the amount of motion for different game designs. Suznjevic *et al.* [23] demonstrated the relation between game genre and network characteristics (e.g. bandwidth usage and packet rate). And finally, Jarschel *et al.* [24] showed the impact of game genre on perceived QoE under the same network conditions.

In addition to game genre, several other factors contribute to the maximum tolerable delay [7]. For example experienced users are more sensitive to delay compared to novice users. In general, for a high QoE, the network delay must be below 100 ms [25].

C. DATA CENTER NETWORK (DCN) AND SOFTWARE DEFINED NETWORKING (SDN)

A cloud data center can host many delay sensitive applications. A study of a DCN consisting of a 6000 production server clusters has shown that it is incapable of handling network flows with tight completion deadlines as a result of the limited capacity of its switches and links [26]. Most DCNs today adopt a multi-tier network architecture that consists of distinct layers of network elements (e.g. core, aggregation and access). Typically, a higher number of layers is expected to result in greater network availability. In a cloud gaming data center, a long packets downstream is transmitted as compressed video. The game flows are delay sensitive and should meet certain delay requirements to provide the user with satisfactory QoE. Moreover, the game flows are also throughput sensitive, bounded by the overall long flow completion time. Given the characteristics of the network flows transmitting within the cloud gaming data center, the congestion in data center switches is inevitable. Congestion control solutions through overprovisioning (OP) are commonly used to reduce the effects of congestion in the DCN. However, OP solutions can be expensive as they require the deployment of additional resources in the DCN. Conversely, current data centers use multipath routing along with traffic engineering to improve efficiency as well as reliability. Modern networks employ multipath routing consisting of the shortest and alternate paths to avoid interruption of service. Even though several approaches exist to determine the optimal paths in the network, they usually select the best routes based on the link metrics provided at setup time, and they often fail to take into account the current status of links, such as current available bandwidth, delay and packet loss. Also, they do not consider the requirements of data flows passing through the network. In traditional networks, despite the fact that a router advertises its routing table to other routers, each router autonomously controls its own routing table using the incomplete network information it possesses. This distributed control is not efficient with increasing traffic. So, SDN presents a new approach for the centralization of network control. Basically, SDN separates the network control and forwarding planes to enable the optimization of each layer independently. Also, SDN brings the application layer and the network layer closer together. SDN allows the network to be more adaptable to different conditions and assists the network to respond to application requests in real time. This agility in control can be effectively exploited by cloud gaming systems. However, current SDN-enabled switches have very narrow capacity limits [27].

The OpenFlow (OF) protocol is the most commonly used control protocol in SDN-enabled networks. It is used by an SDN controller to dictate forwarding rules to switches and routers. The flow table is the principal element of OF. It enables OF switches to cache the network control policies and rules. Due to hardware limitations, most of the current OF switches can store fewer than a thousand entries in their flow tables. Hence, flow table capacity is a resource that must be taken into account during flow control.

Generally speaking, the network resource allocation techniques using the OF protocol can be divided into two main types: 1) network-centric and 2) application-centric. Network-centric methods consider network status in real-time to maximize the network resources utilization [28], [29], while application-centric techniques assess the communicating applications' needs to maximize fairness in allocating network resources [30].

III. PROPOSED METHOD: AHP BASED GAME AWARE ROUTING (AGAR)

We propose AGAR (as depicted in Fig. 1), a bi-objective optimization method that considers network status and game

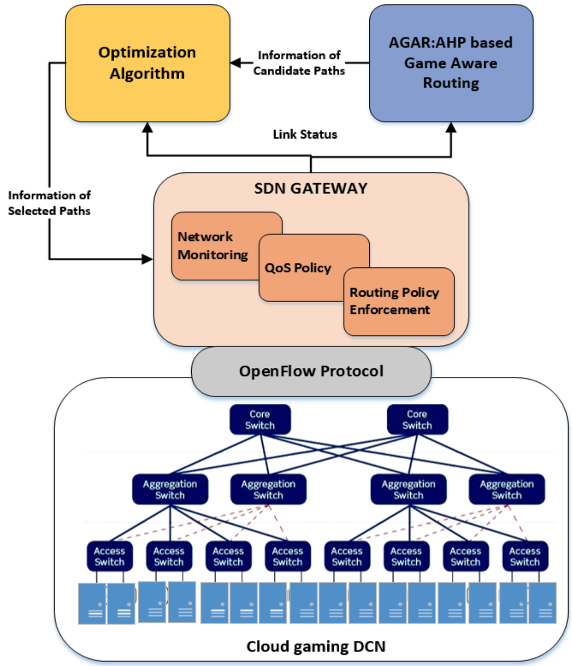


FIGURE 1. AHP based game aware routing framework (AGAR).

characteristics to select the appropriate network path between core and ToR switches for a gaming session request. The proposed optimization scheme, as it will be demonstrated in Section A, cannot be solved in polynomial time for large instances of the problem. Therefore, we employ an iterative approach to relax the original problem and find the optimal solution. We use the AHP metaheuristic method to find the initial solution that we feed into the optimization scheme to select the best routing path in the cloud network. The goal of AGAR is to find the candidate paths that reduce the end-to-end delay experienced by gamers and improve their QoE based on game flow characteristics. Since the optimization scheme uses a global view of the network status as its input, an SDN controller is used to decide on the forwarding strategy for the streams of gaming data within the data center.

Conventional SDN controllers usually find the optimal path in terms of different network criteria (e.g., shortest delay, least hop-count etc.) [6], [31], [32]. Conversely, in our proposed method, first, all possible paths from the requesting core switch to the ToR switches are identified by the controller. While this is an NP-problem, the graphs constructed based on the current architectures of data centers (e.g. Fat-tree [33] and VL2 [34]) are simple, and the source (i.e. core switch) is fixed for all flows. Therefore, the problem of finding all possible paths can be solved using the algorithm described in [6]. Afterwards, the AHP method is utilized to find the candidate solution that will feed into the optimization scheme to find the optimal routing path for a gaming session request.

A. PROBLEM FORMULATION

The network topology of a data center can be modeled by a graph G(V,P), in which V indicates the set of nodes mapped to network (core, aggregation and access) switches, and P denotes the set of disjoint paths linking core switches to the game servers. Provided that there are M active gaming sessions in the data center, the index i ($1 \leq i \leq M$) represents i^{th} gaming session. Let j be the number of paths between the core switch and ToR where $1 \leq j \leq N$. Let φ_{ij} denote the network delay experienced by game session i if its traffic is routed through path j. the first objective function aims to find the network paths that minimize the network delay experienced by the gaming session, so we define f_1 as follows:

$$
\begin{aligned}
f_1 &= \sum_{i=1}^{M}\sum_{j=1}^{N}\varphi_{ij}x_{ij}\\
&ST:\\
&\qquad i.\ \sum_{j=1}^{N} x_{ij} = 1, \quad \forall i \in \{1,\ldots,M\}\\
&\qquad ii.\ \sum_{j=1}^{N} \varphi_{ij}x_{ij} \leq \varphi_{max}, \forall i \in \{1,\ldots,M\}\\
&\qquad iii.\ x_{ij} \in \{0,1\} \quad \forall i \in \{1,\ldots,M\},\\
&\qquad\qquad \forall j \in \{1,\ldots,N\}
\end{aligned}
\tag{1}
$$

In (1), we define a binary decision variable x_{ij} that is equal to 1 if game session i is served by the connecting path j, and 0 otherwise (Constraint iii).

Only, a single path is assigned to each gaming session, and it is guaranteed by Constraint i, and the total amount of network delay is confined by constraint ii to a predefined delay (φ_{max}), i.e. a maximum tolerable delay within the data center as explained in [7].

φ_{max} depends on the genres of games executing within the data center. In addition to the network delay experienced by gaming sessions, we define the function f_2 to express the bandwidth resource utilization within the data center network.

Given that traffic flow of gaming session i is routed through the path j, the fraction of available bandwidth in path j dedicated to gaming session i is represented by β_{ij}. Hence, the bandwidth allocated to gaming session i using path j can be denoted as $\beta_{ij}x_{ij}$ where x_{ij} is a binary variable, as explained above.

Table I presents all notations used throughout this article. For each gaming session, a utility function (denoted by U_i for i^{th} gamer) is defined to capture the level of satisfaction of i^{th} gamer from receiving the bandwidth β_{ij}. It should be noted that this satisfaction corresponds to the quality of gaming perceived by the gamer.

In our previous work [35], we derived the utility functions of four popular game genres: FPS, sport/Role-Playing Game (RPG), RTS, and ARPG. These utility functions model the relationship between bit rate and objective QoE measured by two metrics: Peak Signal to Noise Ratio (PSNR) and Structural Similarity (SSIM) index.

TABLE 1. Table of notations.

Notation	Description	Notation	Description
M	number of gaming sessions	N	number of available paths
i	Gaming session index	j	Gaming server index
j	Available paths index	φ_{max}	Maximum tolerable delay
x_{ij}	binary variable for selection of a path	U_i	utility function of gaming session i
β_{ij}	Amount of bandwidth assigned to game i	δ_{max}	maximum bandwidth capacity
p	Set of n possible disjoint paths	P_d	Network resource importance parameters for delay
P_b	Network resource importance parameters for bandwidth	b^{δ}	Set of residual bandwidth for each path
d^{φ}	Set of the inverse mean delay for each path	x_{ij}	binary variable for selection of a path
CI_b	Consistency Index	ρ_b	Preference Degree Index for bandwidth
ρ_d	Preference Degree Index for delay	M_{φ}^{d}	pairwise N × N comparison matrices for residual delay
M_{δ}^{b}	pairwise N × N comparison matrices for residual bandwidth	φ_{ij}	network delay experienced by game session i

Therefore, as we properly explained in [45], the utility function of the proposed optimization problem can be obtained by summing up all of gamers' utility functions as follows.

$$\begin{aligned} f_2 &= \sum_{i=1}^{M} U_i(\sum_{j=1}^{N} \beta_{ij}x_{ij}) \\ ST &: \\ &\sum_{i=1}^{M} \beta_{ij}x_{ij} \le \delta_{max} \quad \forall j \in \{1,\ldots,N\} \\ &\sum_{j=1}^{N} x_{ij} = 1, \quad \forall i \in \{1,\ldots,M\} \\ &x_{ij} \in \{0,1\}, \quad \forall j \in \{1,\ldots,N\}, \ \forall i \in \{1,\ldots,M\} \end{aligned} \tag{2}$$

The constraints ii and iii were described previously. Constraint i ensures that the total amount of bandwidth of the gaming sessions on a path j cannot exceed its maximum bandwidth capacity δ_{max}.

In addition to maximizing the overall bandwidth utilization, to prevent any game session from being subjected to starvation, we incorporate some level of fairness among the sessions by applying the logarithmic utility function. So we rewrite the objective function (2) as:

$$f_2 = \sum_{i=1}^{M} \log(\sum_{j=1}^{N} \beta_{ij}x_{ij}) \tag{3}$$

In this route selection problem within the data center, two objectives are simultaneously pursued, (1) to minimize the network delay experienced by the gaming session, and (2) to maximize the bandwidth utilization. Since the objective functions, f_1 and f_2, are not completely compatible, we introduce the optimization problem Z as follows:

$$\begin{aligned} Z &: Min\{f_1\},\ Maxf_2 \equiv Minf_1, -f_2 \\ ST &: \\ &\sum_{j=1}^{N} x_{ij} = 1, \quad \forall i \in \{1,\ldots,M\} \\ &\sum_{i=1}^{M} \delta_{ij}x_{ij} \le \delta_{max}, \quad \forall j \in \{1,\ldots,N\} \\ &\sum_{j=1}^{N} \varphi_{ij}x_{ij} \le \varphi_{max}, \quad \forall i \in \{1,\ldots,M\} \\ &x_{ij} \in \{0,1\} \quad \forall i \in \{1,\ldots,M\},\ \forall j \in \{1,\ldots,N\} \end{aligned} \tag{4}$$

Decision variable x_{ij} turns our problem into an Integer Programming (IP) model. These problems are harder to solve compared to Linear Programming (LP) models because of their intrinsically combinatorial nature. In addition, large problem sizes make IP-based models computationally complex to solve. However, given constraints ii and iii, we can consider that Z consists of two separate Generalized Assignment Problems (GAP) that can be solved in proportional time. We use a standard branch-and-bound algorithm to solve for Z which requires the establishment of the lower bound solution. Although the lower bound can be obtained by the relaxed model of Z, the two sub problems f_1 and f_2 contain binary variables that still makes them computationally costly. Therefore, for the problem Z, the computational complexity of iterative methods (e.g. subgradient) may be excessively high. More importantly, the tuning process to determine the sequence of step-lengths to update successive iterations is not a trivial task. Hence, we use the AHP metaheuristic to define the initial lower bound solution. A branch and bound method is used to ascend from the lower bound to an optimal solution of the generalized problem. In the next section we demonstrate the details of this approach.

B. ANALYTIC HIERARCHY PROCESS (AHP)

The AHP is a structured method to deal with multi-criteria decisions. While a mix of qualitative, quantitative, and sometimes conflicting factors are taken into account to find the best solution among several alternatives, AHP enables the decision maker to find one that best meets problem requirements. Although various multi- criteria models, such as TOPSIS, grey rational analysis, and PROMETHEE II have been proposed to assess, prioritize, rank, and evaluate decision choices, most of these methods are inherently complex when factors with diverse and conflicting priorities are considered. The main difference between AHP and the other decision-making methods is that the factors are weighted to indicate that they are of different importance.

AHP splits a complex problem into a multi-leveled hierarchical process. The number of levels in the hierarchy is used to derive ratio-scaled measures for decision alternatives [36]. AHP generally includes four steps: First, define the problem and state the goal or objective. Second, define the criteria or factors that influence the goal and structure these factors into levels and sublevels. Third, perform paired comparisons between factors using a comparison matrix containing calculated weights, ranked eigenvalues, and consistency measures. Finally, synthesize the ranks of alternatives until the final choice is made. In the following paragraphs, we detail how we applied these steps in the formulation of our proposed AHP method.

1) STEP 1. PROBLEM STATEMENT

As it can be seen in III-A, the DCN is controlled by the SDN, and the OF protocol is used to make our proposed method adaptable to a variety of networks. In our method, the importance of network resources for a particular game genre is specified in the AHP quantitatively. Having these quantized values of network resources, the best path between alternative paths is selected and the results are fed into the optimization scheme of Section 3.1. We adopt the fat-tree 3-tier DCN architecture as our network topology.

In order to describe the algorithmic details of the proposed method, we first introduce some notations:

1) $G(V, E)$: Network topology digraph of a data center where V denotes the set of switches (including core, aggregation and ToR switches) and Edenotes a set of links in the network.
2) $p = \{p_1, p_2, \ldots, p_n\}$: Set of n possible disjoint paths fromV_s: Source Vertex to V_d: Destination Vertex
3) P_d and P_b: Network resource importance parameters for a game genre corresponding to delay sensitivity and bandwidth requirement respectively.
4) $b^\delta = \{b_1^\delta, b_2^\delta, \ldots, b_n^\delta\}$: Set of residual bandwidth for each path. We define residual bandwidth b_{i}^δ as the lowest available bandwidth on any of the links that belong to p_i.
5) $d^\varphi = \{d_1^\varphi, d_2^\varphi, \ldots, d_n^\varphi\}$: Set of the inverse mean delay for each path (i.e., d_{i}^φ corresponds to the inverse of the mean delay on p_i).

2) STEP 2. DEFINE THE CRITERIA OR FACTORS THAT INFLUENCE THE GOAL

In order to build the routing method, the most important factors that influence network resource management have to be specified. In this work, we categorize these factors into network and application factors.

For the network factors, we consider residual bandwidth and delay on the network links. For the application factors, we consider the network resource requirements for different game genres. Game genres define games in terms of having a common style or set of characteristics, e.g. as defined in terms of perspective, gameplay, interaction, objective, etc. Authors in [22] show that the extents of motion and scene complexity vary noticeably among different game genres. A complex scene demands a higher amount of bandwidth compared to a scene with lower complexity. On the other hand, games with higher amounts of motion are more sensitive to delay variation. Accordingly, we summarize the network resource importance for various game genres in Table II.

TABLE 2. Game network resource importance.

GAME GENRE	PERSPECTIVE	NETWORK RESOURCE IMPORTANCE	
		DELAY SENSITIVITY (ρ_d)	BANDWIDTH REQUIREMENT (ρ_b)
First Person Shooter (FPS)/ Action Role-Playing Game/Action-adventure	First	High	Low
Action-Adventure/Action Role-Playing Game (ARPG)	Isometric	Medium-High	Low-Medium
Third Person Shooter/Role-Playing Game (RPG)	Linear	Low-Medium	Low-Medium
Trading Card Game/Real Time Strategy (RTS)	Omnipresent	Medium	Medium-High

We then design Table III, which maps the network resource importance qualifications used in Table II to discrete numeric values that can be employed in the AHP based decision making.

TABLE 3. Rating scale for importance index.

DEGREE OF IMPORTANCE	DEFINITION
1	Very High
0.8	High
0.6	Medium
0.4	Low
0.2	Very Low
0.1,0.3,0.5,0.7	Intermediate Values between the two adjacent degrees

3) STEP 3. PAIRWISE COMPARISON MATRIX (PCM)

1) In AHP, the Pairwise Comparison Matrix (PCM) is used to calculate the priority vector of alternatives [36]. Having two main criteria, bandwidth and delay, we build two comparison matrices for the set of N candidate paths. Therefore, we define M_δ^b and M_φ^d as the pairwise $N \times N$ comparison matrices for residual bandwidth and inverse mean delay. The matrices are

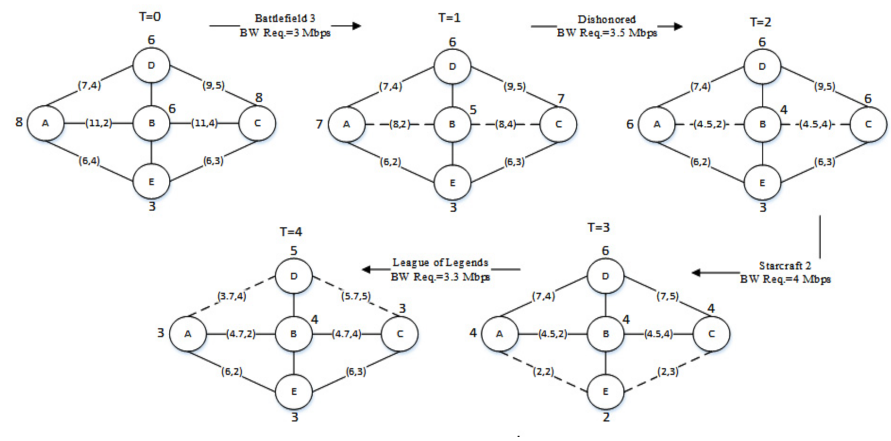


FIGURE 2. Toy Network Example.

obtained as follows:

$$(M_\delta^b)_{jk} = b_j^\delta / b_k^\delta, (M_\varphi^d)_{jk} = d_j^\varphi / d_k^\varphi \tag{5}$$

$$M_\delta^b = \begin{pmatrix} 1 & b_1^\delta/b_2^\delta & b_1^\delta/b_3^\delta & \dots & b_1^\delta/b_N^\delta \\ b_2^\delta/b_1^\delta & 1 & b_2^\delta/b_3^\delta & \dots & b_2^\delta/b_N^\delta \\ b_3^\delta/b_1^\delta & b_3^\delta/b_2^\delta & 1 & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_N^\delta/b_1^\delta & b_N^\delta/b_2^\delta & b_N^\delta/b_3^\delta & \dots & 1 \end{pmatrix} \tag{6}$$

$$M_\mu^d = \begin{pmatrix} 1 & d_1^\varphi/d_2^\varphi & d_1^\varphi/d_3^\varphi & \dots & d_1^\varphi/d_N^{\delta\varphi} \\ d_2^\varphi/d_1^\varphi & 1 & d_2^\varphi/d_3^\varphi & \dots & d_2^\varphi/d_N^\varphi \\ d_3^\delta/d_1^\delta & d_3^\delta/d_2^\delta & 1 & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_N^\varphi/d_1^\varphi & d_N^\varphi/d_2^\varphi & d_N^\varphi/d_3^\varphi & \dots & 1 \end{pmatrix} \tag{7}$$

We calculate the priority vectors of matrices M_δ^b and M_φ^d which correspond to the normalized Eigenvector of each one of them. We define ω_b and ω_d as the normalized Eigenvectors and λ_b, λ_d as the Eigenvalues of M_δ^b and M_μ^d respectively.

2) We check the consistency of the pairwise comparison matrices using the test if $\lambda_{b_{max}} = N$, then the pairwise comparison matrix M_δ^b is perfectly consistent. We calculate the Consistency Index (CI) and the Consistency Ratio (CR) to measure the consistency degree for two comparison metrics as follows [36]:

$$CI_b = \frac{\lambda_{b_{max}} - N}{N-1}, \quad CI_d = \frac{\lambda_{d_{max}} - N}{N-1} \tag{8}$$

$$CR_b = \frac{CI_b}{RI}, \quad CR_d = \frac{CI_d}{RI} \tag{9}$$

where RI is a Random Index scale. In this work, we use the RI scaled proposed by [37] as displayed in Table IV. The pairwise comparison metrics are accepted if the value of CR is smaller or equal to 0.1.

TABLE 4. Random consistency values RI(N).

N	3	4	5	...	13	14	15
RI	0.52	.0.881	1.108		1.5551	1.5713	1.5838
	45	5	6				

3) We define the game network resource importance indices ρ_b *and* ρ_d as the weights corresponding to the importance of bandwidth and delay, respectively, for a particular game genre. The importance of bandwidth and delay for a game genre is obtained from Tables II and III. The values of ρ_b, ρ_d should satisfy the following constraint:

$$\rho_b + \rho_d = 1 \tag{10}$$

where:

$$0 \le \rho_b \le 1, \quad 0 \le \rho_d \le 1 \tag{11}$$

Using the priority vectors for the main network resource criteria, ω_b, ω_d and the game network resource importance indices ρ_b, ρ_d, we calculate the Preference Degree Index (PDI) for a candidate path as follows:

$$PDI_i = \rho_b * \omega_b + \rho_d * \omega_d \tag{12}$$

The PDI is a vector with N elements, each one corresponding to the preference of a candidate path. The path with the highest preference value is chosen as a routing path for the requested game flow. The entire process of AHP is summarized in Algorithm 1.

In order to further illustrate the inner workings of AHP, we present an example that details the process of route selection. Fig. II shows a toy data center consisting of five

Algorithm 1 Analytical Hierarchy Process

1:
Inputs:
G(V, E) : Data Center Topology Digraph
$\mathbf{v_s}$ **: Source Vertex**
$\mathbf{v_d}$ **: Destination Vertex**
Output: **Routing path from** $\mathbf{v_s}$ **and** $\mathbf{v_d}$
2: $p = (G, W, v_s, v_d)$
3: Build the PCMs M^b_δ, M^d_φ
4: Calculate the normalized Eigenvectors and Eigen values for each matrix
Set ω_b, ω_d// as Eigenvectors
5: Measure the consistency ratio (CR) of the PCMs
6: If CR > 0.1 then
go to step 3
8: Establish the game network resource importance indices P_b, P_d for the requested game genere
7: Calculate the PDI for each candidate path i

$$PDI_i = P_b * \omega_b + P_d * \omega_d$$

9: Set the route with maximum preference value in the PDI vector as the routing path

OF-enabled switches. The flow entry capacity of each switch is marked next to it. Also, the pair of bandwidth and delay for each connecting link is presented using the format (b, d), where b corresponds to bandwidth and d corresponds to delay. We assume that there are four game flow requests from games belonging to diverse genres, namely FPS, RPG, ARPG, and RTS, and three available routing paths with different characteristics. Note the game flow requests arrive in the same order as they are stated in the latter statement. Using Tables 1 and 2 and the constraints of Equations (9) and (10), we set the game network resource importance indices ρ_b, ρ_d for FPS, ARPG, RPG, and RTS games as (0.147, 0.853), (0.643, 0.357), (0.29, 0.71) and (0.65, 0.345), respectively. The route section results are presented in Table V, where we can see that the flow pertaining to the game with high delay sensivity is routed through the path with the lower delay. Moreover, the game with the high bandwidth requirement is more likely to be assigned to the path with more residual bandwidth.

TABLE 5. Route selection results.

Game Genre	Tested game	Bandwidth Requirement	PDI (Path1)	PDI (Path2)	PDI (Path3)	Selected Route
FPS	Battlefield 3	3	0.28	0.373	0.34	ABC
RPG	Dishonored	3.5	0.316	0.367	0.292	ABC
ARPG	Starcraft 2	4	0.312	0.334	0.353	AEC
RTS	League of Legends	3.3	0.433	0.342	0.224	ADC

IV. PERFORMANCE EVALUATION

A. EXPERIMENTAL SETUP

In this section, we evaluate the performance of the proposed routing scheme. Our simulations are run on an Ubuntu version 14.4 (3.4 GHz Intel workstation) with 8 GB of RAM and the coding was developed using MATLAB version 8.4. In order to simulate the DCN architecture, we use a Mininet emulator on the Oracle virtual machine version 4.3. The Mininet emulator enables us to create a realistic network experiment with OF and SDN. We implement a fat-tree [33] based architecture where a collection of switches and servers are built within a pod. In the fat-tree architecture, the numbers of required switches as well as servers in each pod are dependent on the number of ports in each switch. For example, if each switch has k ports, the DCN consists of k pods and each pod has $k/2$ access and aggregation switches, and $k^2/4$ core switches and servers. The DCN is controlled by an OF SDN controller deployed using the POX controller libraries [38]. The application running on the controller manages network flows and informs the open virtual switches (vSwitches) where to send the packets. In the experiments, we consider a network topology that consists of 20 vSwitches and 600 links. The OF table capacity of each vSwitch is 1000 and the maximum bandwidth of each link is 20 Mbps. While the number of switches and links are fixed, we study two scenarios with varying number of game flows running within the DCN: Scenario 1 with 100 game flows and Scenario 2 with 200 games flows. The game flows in both scenarios belong to four games: Battlefield 3, Dishonored, Starcraft 2 and League of Legends. These games belong to the following game genres respectively: FPS, Action-adventure, RTS, and Multiplayer online battle arena and have varying delay and bandwidth requirements. The details of the network traffic parameters for these games are presented in Table VI [19]. The flow requests arrive at random intervals (normal distribution) and are then assigned to a route by the SDN controller. In our simulation, we compare AGAR with the Equal Cost Multi-Path routing (ECMP) [39], [40], Hedera [41] and the Delay-Based Dijkstra (DBD) multi-path routing methods in data centers. For ECMP, the switch identifies routing path locally using a hash algorithm. In fact, flows distribute with an equal probability across all paths. On the other hand, for DBD, we employ the algorithm proposed in [6] in which routes with the least delay can be obtained by applying Dijkstra's algorithm and using delay on the links as the cost. And finally, for Hedera, a centralized scheduler collects flow information from the edge switches, performs scheduling every few seconds, and distributes flows.

B. COMPARISON METHOD

In order to compare the proposed algorithm with other candidate methods, we employ the Kullback-Leibler divergence (KL-divergence) method so-called related entropy [42]. The KL divergence emanates from the field of information theory, and it is now accepted widely as a good measure

TABLE 6. Network traffic parameters of games.

Games	Genre	Upstream traffic		Downstream traffic	
		Mean Packet Size (bytes)	Mean Inter- Departure time (ms)	Mean Packet Size (bytes)	Mean of Inter-Departure time (ms)
Battlefield 3	First Person Shooter (FPS)	35.03	2.9	1,108.59	1.6
Dishonored	Action-adventure	61.19	3.3	722.58	1.8
Starcraft 2	Real Time Strategy (RTS)	34.01	2.6	955.65	2.48
League of Legends	Multiplayer online battle arena	59.83	1.5	674.38	2.1

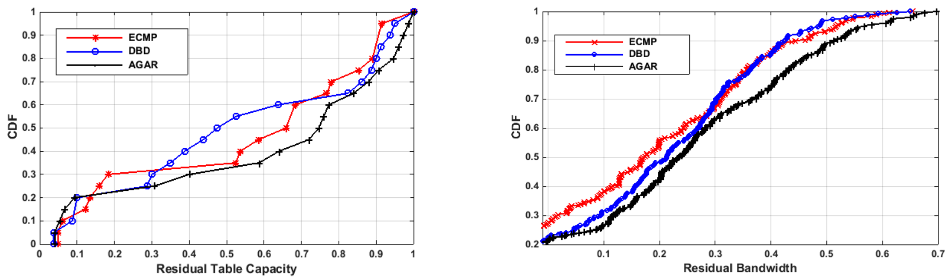


FIGURE 3. CDF of Residual Flow Table Capacity and Residual Bandwidth, 100 Game flows.

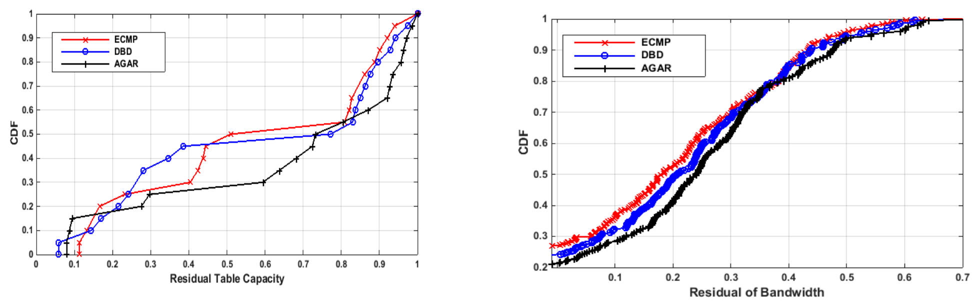
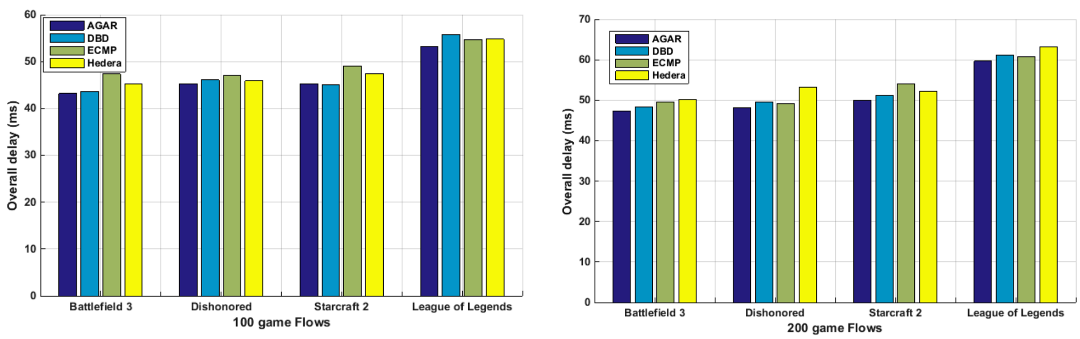


FIGURE 4. CDF of Residual Flow Capacity and Bandwidth, 200 Game flows.



(a) (b)

FIGURE 5. The Overall delay Experienced by Users a) Number of users = 100 b) Number of users = 200.

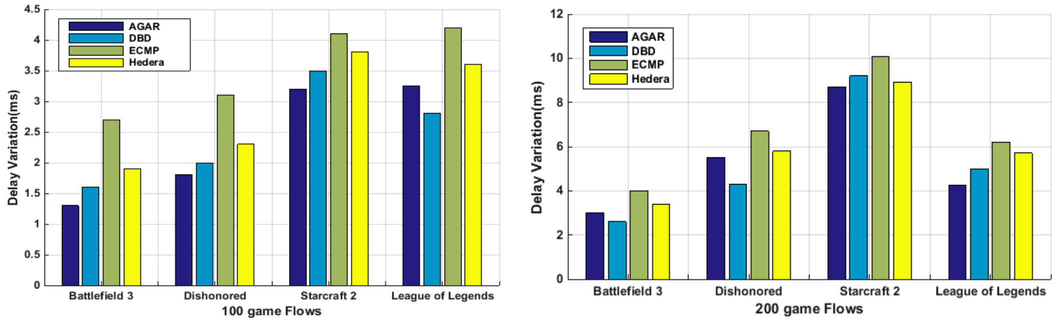
of the difference between two probability distributions as follows [43], [44]:

$$D_{KL}\,(P||Q) = \sum_i P(i) \log P(i)/Q(i) \tag{13}$$

where D_{KL} presents the logarithmic difference between the probabilities P and Q in which P represents the true uniform distribution of results, and Q is the observed data during the experiments and defined as follows:

$$P\,(c_i) = C(i)/\sum_i^n C\,(i)\,, 1 \le i \le n,$$

n indicates number of OF switches (14)



(a) (b)

FIGURE 6. **The Average of delay variation Experienced by Users a) Number of users = 100 b) Number of users = 200.**

$$P(b_i) = b(i) / \sum_{i}^{n} b(i), 1 \leq i \leq n, \quad \text{n indicates available links} \tag{15}$$

Equations (14) and (15) represent the probability of residual flow table capacity and residual bandwidth respectively.

V. PERFORMANCE EVALUATION

We employ KL-divergence as a comparison method between the three evaluated approaches. Hence, in equation (14), P represents a true uniform distribution of network resources across all paths and Q represents the actual distribution achieved by the routing method. Figure III presents the Cumulative Distribution Function (CDF) for the residual flow table capacity and residual bandwidth measured for simulation A (100 game flows). Similarly, Figure IV shows the CDF of the network resources for simulation B (200 game flows). Furthermore, Table VII lists the KL-divergence in terms of bandwidth and flow entry capacity for the evaluated methods.

TABLE 7. **Results of KL- divergence calculation.**

		KL-Divergence	ECMP	DBD	AGAR
A	100 game Flows	Residual of Bandwidth	0.6677	0.6345	0.5719
		Residual of Flow Capacity	0.7707	0.5344	0.3967
B	200 game Flows	Residual of Bandwidth	0.6626	0.6259	0.6062
		Residual of Flow Capacity	0.4144	0.375	0.3425

It can be seen that the KL-divergence value for AGAR is lower compared to ECMP and DBD, and much closer to a uniform distribution of flows across paths. This shows that the game traffic is fairly distributed throughout the network which results in diminishing network congestion.

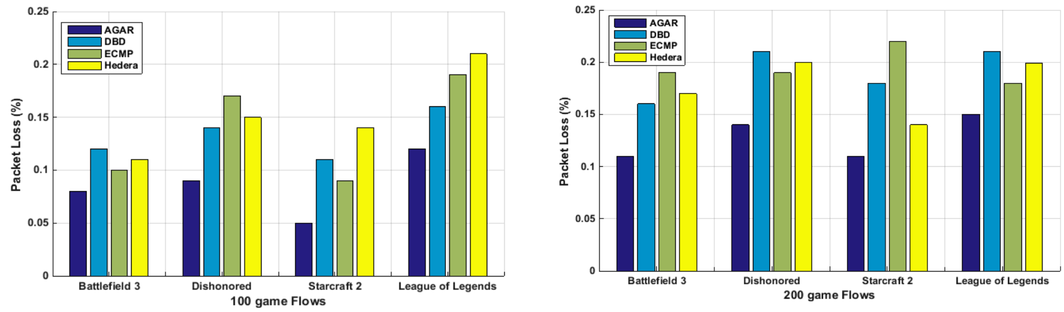
The average delay experienced by the game flows belonging to the different games is depicted in Figures Va and Vb. For Scenario 1, the average delay experienced by players is 1.9%, 5% and 3.7% lower when we use AGAR compared to DBD, ECMP and Hedera, respectively. For Scenario 2, the average delay experienced by players is 2.4%, 2.7% and 9.5% lower when we use AGAR compared to DBD, ECMP and Hedera, respectively.

Figures VIa and VIb show that the average delay variation experienced by players is 14% and 8% lower when AGAR is used compared to ECMP for Scenarios 1 and 2, respectively. As for DBD, the results are almost comparable with AGAR. On average, AGAR decreases the delay variation experienced by users by almost 7% and 2% with respect to DBD for Scenarios 1 and 2 respectively. Also, the average delay variation experienced by players using the proposed method is almost 6% and 4% less compared to Hedera for Scenarios 1 and 2, respectively.

Although AGAR renders lower overall delay variations on the network, there are some cases where DBD performs better for high delay sensitive games as evidenced by the League of Legends and Dishonored results. DBD assigns the path with the minimum delay to a game flow request without accounting for bandwidth. Therefore, this approach might, under limited circumstances, reduce the delay variations experienced by the flows of a high delay sensitive game, at the expense of other games in the network. Aside from the improvements in experienced delay, we compare the Packet Loss Rate (PLR) for different schemes. The results are illustrated in Figures VIIa and VIIb. For Scenario 1, the proposed method achieves almost 14%, 23% and 27% lower packet loss than the DBD, ECMP and Hedera methods, respectively. For Scenario 2, the proposed method continues to outperform DBD, ECMP and Hedera in terms of PLR by 19%, 17% and 14%, respectively.

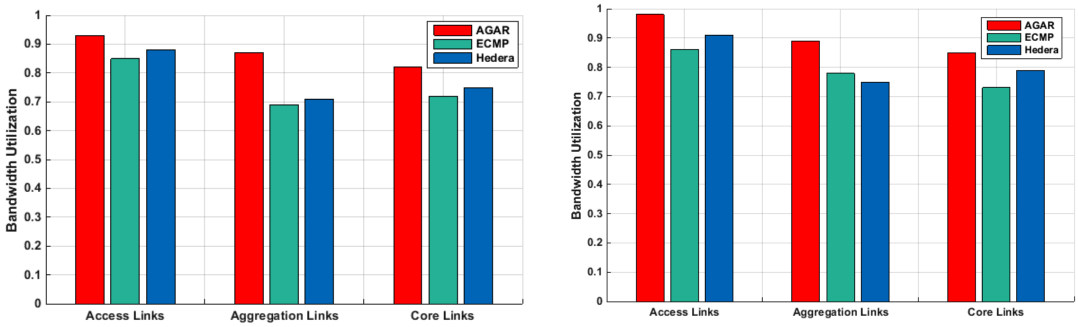
This improvement stems from the fact that the proposed scheme monitors bandwidth usage of network paths, and dynamically regulates traffic load accordingly.

The bandwidth utilized by all network layers (i.e. core, aggregation and access) in scenarios 1 and 2 is illustrated in Figures VIIIa and VIIIb respectively. Our proposed method yields improvement in bandwidth utilization compared to ECMP and Hedera by 28% and 35%, respectively. The effective bandwidth utilization can enhance network fairness and contribute to traffic congestion avoidance. In scenario 2, access link bottlenecks occur, hence, the proposed approach maximizes the utilization of all network links. Our proposed



(a) (b)

FIGURE 7. The Average of Packet Loss rate Experienced by Users a) Number of users = 100 b) Number of users = 200.



(a) (b)

FIGURE 8. The bandwidth utilization of layers for a) Number of users = 100 b) Number of users = 200.

method better utilizes available bandwidth and outperforms the ECMP and Hedera schemes by 5% and 7% respectively.

VI. CONCLUSION

In cloud gaming, the major computational part of game processing is performed on the cloud's data center. Hence, a properly designed DCN can provide high quality games to end-users and reduce cost. In this work, we focus on DCN's resource management using SDN. We present an optimization-based routing scheme that efficiently assigns gaming flows to communication paths while jointly minimizing overall delay and maximizing bandwidth utilization. Unlike the proposed method, existing cloud gaming DCN routing schemes do not consider the QoS characteristics of game genres for path assignment. In our evaluation, we compared the proposed method to three other representative approaches: the ECMP, DBD and Hedera. We conducted two simulations involving 100 and 200 game flows in a fat-tree DCN. Our results indicate that the proposed method results in the reduction of delay within the DCN compared to ECMP, DBD and Hedera by an average of 2.7%, 2.2% and 6.8%, respectively. It should be also noticed that there are no additional packet losses experienced when reducing the delay and jitter; and therefore the QoE perceived by a cloud gamer is being improved considerably.

Also, our results show that the proposed optimization scheme produces a more balanced assignment of game flows across the DCN paths compared to the other three methods. This can potentially help prevent routing failures.

REFERENCES

[1] S. Shirmohammadi, M. Abdallah, D. T. Ahmed, Y. Lu, and A. Snyatkov, "Introduction to the special section on visual computing in the cloud: Cloud gaming and virtualization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 12, pp. 1955–1959, Dec. 2015.

[2] Technology Analysis Business Wire. *Research and Markets: Global Cloud Gaming Market 2015–2020-Gaming Audience, Devices &Amp.* Accessed: Apr. 27, 2017. [Online]. Available: http://www.businesswire.com/news/home/20150708005534/en/Research-Markets-Global-Cloud-Gaming-Market-2015-2020

[3] GeForce. *NVIDIA GeForce GRID—A Glimpse at the Future of Gaming*. Accessed: Apr. 27, 2017. [Online]. Available: http://www.geforce.com/whats-new/articles/geforce-grid

[4] S. Choy, B. Wong, G. Simon, and C. Rosenberg, "A hybrid edge-cloud architecture for reducing on-demand gaming latency," *Multimedia Syst.*, vol. 20, no. 5, pp. 503–519, 2014.

[5] *Gaikai Will be Fee-Free, Utilize 300 Data Centers in the US.* Accessed: Apr. 27, 2017. [Online]. Available: https://www.engadget.com/2010/03/11/gaikai-will-be-fee-free-utilize-300-data-centers-in-the-us/

[6] M. Amiri, H. Al Osman, S. Shirmohammadi, and M. Abdallah, "An SDN controller for delay and jitter reduction in cloud gaming," in *Proc. 23rd ACM Int. Conf. Multimedia-(MM)*, 2015, pp. 1043–1046.

[7] M. Claypool and K. Claypool, "Latency and player actions in online games," *Commun. ACM*, vol. 49, no. 11, p. 40, 2006.

[8] F. Kuipers, T. Korkmaz, M. Krunz, and P. Van Mieghem, "An overview of constraint-based path selection algorithms for QoS routing," *IEEE Commun. Mag.*, vol. 40, no. 12, pp. 50–55, Dec. 2002.

[9] M. Claypool, R. Eg, and K. Raaen, "The effects of delay on game actions," in *Proc. Annu. Symp. Comput.-Hum. Interact. Play Companion Extended Abstracts-CHI PLAY Companion*, 2016, pp. 117–123.

[10] M. Claypool and K. Claypool, "Latency can kill: Precision and deadline in online games," in *Proc. 1st Annu. ACM SIGMM*, 2010, pp. 215–222.

[11] S. Choy, B. Wong, G. Simon, and C. Rosenberg, "The brewing storm in cloud gaming: A measurement study on cloud to end-user latency," in *Proc. Annu. Workshop Netw. Syst. Support Games*, 2012, p. 2.

[12] S. Shi, C.-H. Hsu, K. Nahrstedt, and R. Campbell, "Using graphics rendering contexts to enhance the real-time video coding for mobile cloud gaming," in *Proc. 19th ACM Int. Conf. Multimedia-(MM)*, 2011, p. 103.

[13] V. Jalaparti, M. Caesar, S. Lee, J. Pang, and J. Van Der Merwe, "SMOG: A cloud platform for seamless wide area migration of online games," in *Proc. 11th Annu. Workshop Netw. Syst. Support Games*, 2012, p. 9.

[14] H.-J. Hong, D.-Y. Chen, C.-Y. Huang, K.-T. Chen, and C.-H. Hsu, "Placing virtual machines to optimize cloud gaming experience," *IEEE Trans. Cloud Comput.*, vol. 3, no. 1, pp. 42–53, Jan. 2015.

[15] P. B. Beskow, P. Halvorsen, and C. Griwodz, "Latency reduction in massively multi-player online games by partial migration of game state," in *Proc. 2nd Int. Conf. Internet Technol. Appl.*, 2007, pp. 153–163.

[16] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, "The nature of data center traffic," in *Proc. 9th ACM SIGCOMM Conf. Internet Meas. Conf. (IMC)*, 2009, p. 202.

[17] M. Amiri, H. Al Osman, S. Shirmohammadi, and M. Abdallah, "SDN-based game-aware network management for cloud gaming," in *Proc. Annu. Workshop Netw. Syst. Support Games*, 2015, pp. 1–6.

[18] M. Claypool, D. Finkel, A. Grant, and M. Solano, "Thin to win? Network performance analysis of the OnLive thin client game system," in *Proc. 11th Annu. Workshop Netw. Syst. Support Games (NetGames)*, 2012, pp. 1–6.

[19] M. Manzano, M. Urueña, M. Sužnjević, E. Calle, J. A. Hernández, and M. Matijasevic, "Dissecting the protocol and network traffic of the OnLive cloud gaming platform," *Multimedia Syst.*, vol. 20, no. 5, pp. 451–470, 2014.

[20] C.-Y. Huang, C.-H. Hsu, Y.-C. Chang, and K.-T. Chen, "GamingAnywhere," in *Proc. 4th ACM Multimedia Syst. Conf. (MMSys)*, 2013, pp. 36–47.

[21] M. Suznjevic, L. Skorin-Kapov, and M. Matijasevic, "The impact of user, system, and context factors on gaming QoE: A case study involving MMORPGs," in *Proc. Annu. Workshop Netw. Syst. Support Games*, 2013, pp. 1–6.

[22] M. Claypool, "Motion and scene complexity for streaming video games," in *Proc. 4th Int. Conf. Found. Digit. Games (FDG)*, 2009, p. 34.

[23] M. Suznjevic, J. Beyer, L. Skorin-Kapov, S. Moller, and N. Sorsa, "Towards understanding the relationship between game type and network traffic for cloud gaming," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jun. 2014, pp. 1–6.

[24] M. Jarschel, D. Schlosser, S. Scheuring, and T. Hoãÿfeld, "An evaluation of QoE in cloud gaming based on subjective tests," in *Proc. 5th Int. Conf. Innov. Mobile Internet Services Ubiquitous Comput.*, 2011, pp. 330–335.

[25] A. Beznosyk, P. Quax, K. Coninx, and W. Lamotte, "Influence of network delay and jitter on cooperation in multiplayer games," in *Proc. 10th Int. Conf. Virtual Reality Continuum Appl. Ind. (VRCAI)*, 2011, p. 351.

[26] M. Alizadeh *et al.*, "Data center TCP (DCTCP)," in *Proc. ACM SIGCOMM Conf. (SIGCOMM)*, 2010, vol. 40. no. 4, p. 63.

[27] B. Stephens, A. Cox, W. Felter, C. Dixon, and J. Carter, "PAST," in *Proc. 8th Int. Conf. Emerg. Netw. Experim. Technol. (CoNEXT)*, 2012, p. 49.

[28] D. Kuptsov, B. Nechaev, and A. Gurtov, "Demand-aware flow allocation in data center networks," in *Proc. ACM Workshop Capacity Sharing (CSWS)*, 2012, p. 53.

[29] J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, A. R. Curtis, and S. Banerjee, "DevoFlow," in *Proc. 9th ACM SIGCOMM Workshop Hot Topics Netw. (Hotnets)*, 2010, pp. 1–6.

[30] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race, "Towards network-wide QoE fairness using openflow-assisted adaptive video streaming," in *Proc. ACM SIGCOMM Workshop Future Hum.-Centric Multimedia Netw. (FhMN)*, 2013, p. 15.

[31] R. Wang, D. Butnariu, and J. Rexford, "OpenFlow-based server load balancing gone wild," in *Proc. Workshop Hot-ICE*, vol. 11, Mar. 2011. [Online]. Available: https://www.usenix.org/legacy/event/hotice11/tech/full_papers/Wang_Richard.pdf

[32] Asia-Pacific Signal and Information Processing Association. Annual Summit and Conference, "OpenQoS: An OpenFlow controller design for multimedia delivery with end-to-end Quality of Service over," in *2012 Conference Handbook: Asia-Pacific Signal and Information Processing Association Annual Summit and Conference: December 3-6, 2012, Hollywood, California, USA*. Asia-Pacific Signal and Information Processing Association, 2012 Annual Summit and Conference International Organizing Committee, 2012.

[33] M. Al-Fares, A. Loukissas, A. Vahdat, M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *Proc. ACM SIGCOMM Conf. Data Commun. (SIGCOMM)*, 2008, vol. 38. no. 4, p. 63.

[34] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, p. 68, Dec. 2008.

[35] M. Amiri, H. Al Osman, S. Shirmohammadi, and M. Abdallah, "Toward delay-efficient game-aware data centers for cloud gaming," *ACM Trans. Multimed. Comput. Commun. Appl.*, vol. 12, no. 5s, pp. 1–19, Sep. 2016.

[36] T. L. Saaty, "How to make a decision: The analytic hierarchy process," *Eur. J. Oper. Res.*, vol. 48, no. 1, pp. 9–26, Sep. 1990.

[37] J. A. Alonso and M. T. Lamata, "Consistency in the analytic hierarchy process: A new approach," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 14, no. 4, pp. 445–459, Aug. 2006.

[38] *POX Wiki—Open Networking Lab—Confluence*. Accessed: Apr. 27, 2017. [Online]. Available: https://openflow.stanford.edu/display/ONL/POX+Wiki

[39] A. Greenberg *et al.*, "VL2: A scalable and flexible data center network," in *Proc. ACM SIGCOMM Conf. Data Commun. (SIGCOMM)*, vol. 39, no. 4, p. 51, 2009.

[40] K. He, E. Rozner, K. Agarwal, W. Felter, J. Carter, and A. Akella, "Presto: Edge-based load balancing for fast datacenter networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 465–478, 2015.

[41] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, "Hedera: Dynamic flow scheduling for data center networks," in *Proc. NSDI*, vol. 10. 2010, p. 19.

[42] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Stat.*, vol. 22, no. 1, pp. 79–86, 1951.

[43] T. Homem-de-Mello, "A study on the cross-entropy method for rare-event probability estimation," *INFORMS J. Comput.*, vol. 19, no. 3, pp. 381–394, Aug. 2007.

[44] R. Y. Rubinstein, "Cross-entropy and rare events for maximal cut and partition problems," *ACM Trans. Model. Comput. Simul.*, vol. 12, no. 1, pp. 27–53, Jan. 2002.

[45] M. Amiri, H. Al Osman, and S. Shirmohammadi, "Datacenter traffic shaping for delay reduction in cloud gaming," in *Proc. IEEE Int. Symp. Multimedia (ISM)*, Dec. 2016, pp. 569–574.

MARYAM AMIRI is currently pursuing the Ph.D. degree with the Department of Electrical Engineering and Computer Science, Ottawa University, ON, Canada. She is currently a Research Assistant with the Discover Laboratory, School of Electrical Engineering and Computer Science, University of Ottawa. She has deep expertise in the networking, software defined networking, network function virtualization, data center network, and network support for multimedia applications. She is also involved in route optimizing in data center for cloud gaming and published several publications to her name and specializes in this area.

ASHKAN SOBHANI received the B.Sc. degree in computer architecture design from the University of Science and Technology, Tehran, Iran, in 2003, the M.A.Sc. degree in computer engineering from Tehran University, in 2006, and the Ph.D. degree in electrical and computer engineering from the University of Ottawa, in 2017. His current research interests include optimization in multimedia systems and networks, in particular, dynamic adaptive streaming over HTTP, and multi-view video transmission system, and optimizing service delivery over wireless networks. Over the course of his academic journey, he has received several scholarships and awards including the Queen Elizabeth II Graduate Scholarship and the University of Ottawa National Excellence Scholarship.

HUSSEIN AL OSMAN (M'12) received the B.A.Sc. *(summa cum laude)* degree (Hons.) in computer engineering and the M.A.Sc. and the Ph.D. degree in electrical engineering in electrical engineering from the University of Ottawa, in 2007, 2007, and 2014, respectively. He is currently an Assistant Professor with the School of Electrical Engineering and Computer Science, University of Ottawa. He has authored/co-authored over 30 research articles. His current research interests include cloud gaming, software defined networking, affective computing, and human–computer interaction. He is a member of the ACM. Over the course of his academic journey, he has received several scholarships and awards including the 3X NSERC Scholarships, the Queen Elizabeth II Graduate Scholarship, the Best Paper Award from the DS-RT 2008 Conference, and the Part Time Professor Award.

SHERVIN SHIRMOHAMMADI (M'04–SM'04–F'17) received the Ph.D. degree in electrical engineering from the University of Ottawa, Canada. He is currently a Professor with the School of Electrical Engineering and Computer Science, University of Ottawa. He is also the Director of the Distributed and Collaborative Virtual Environment Research Laboratory, where he is involved in research in multimedia systems and networks, specifically video systems, gaming systems, and multimedia-assisted healthcare systems. The results of his research, funded by over U.S. 13 million from public and private sectors, have led to over 300 publications, over 65 researchers trained at the master's, Ph.D., and post-doctoral levels, over 20 patents and technology transfers to the private sector, and a number of awards. He was an Associate Editor of the Springer's *Journal of Multimedia Tools and Applications* from 2004 to 2012. He is currently the Editor-in-Chief of the IEEE Transactions on Instrumentation and Measurement, an Associate Editor of the *ACM Transactions on Multimedia Computing, Communications, and Applications*, an Associate Editor of the IEEE Transactions on Circuits and Systems for Video Technology.

Dr. Shirmohammadi is an IEEE Fellow for contributions to multimedia systems and network measurements, a Lifetime Senior Member of the ACM, a University of Ottawa Gold Medalist, and a licensed Professional Engineer in ON, Canada. He is a Key Member of the IEEE MMTC Multimedia Cloud Computing Interest Group, a member of the IEEE CS Technical Committee on Multimedia Computing, and a member of the IEEE ComSoc Technical Committee on Multimedia Communications.