

Research Article

VR-CPES: A Novel Cyber-Physical Education Systems for Interactive VR Services Based on a Mobile Platform

Hanjin Kim , Heonyeop Shin , Hyeong-su Kim , and Won-Tae Kim 

Smart CPS Lab, The Department of Computer Science & Engineering, KOREATECH University, Cheonan, Republic of Korea

Correspondence should be addressed to Won-Tae Kim; wtkim@koreatech.ac.kr

Received 11 May 2018; Revised 7 July 2018; Accepted 6 August 2018; Published 12 September 2018

Academic Editor: Byeong-Seok Shin

Copyright © 2018 Hanjin Kim et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The evolution of virtual reality technology allows users to immerse themselves into virtual environments, providing a new experience that is impossible in the real world. The appearance of cyber-physical systems and the Internet of things makes humans to understand and control the real world in detail. The integration of virtual reality into cyber-physical systems and the Internet of things may induce innovative education services in the near future. In this paper, we propose a novel, a virtual reality-based cyber-physical education system for efficient education in a virtual reality on a mobile platform, called VR-CPES. VR-CPES can integrate the real world into virtual reality using cyber-physical systems technology, especially using digital twin. We extract essential service requirements of VR-CPES in terms of delay time in the virtual reality service layer. In order to satisfy the requirements of the network layer, we design a new, real-time network technology interworking software, defined as network and time-sensitive network. A gateway function for the interworking is developed to make protocol level transparency. In addition, a path selection algorithm is proposed to make flexible flow between physical things and cyber things. Finally, a simulation study will be conducted to validate the functionalities and performance in terms of packet loss and delay as defined in the requirements.

1. Introduction

Internet of things (IoT) is a new paradigm that plugs things into the Internet in order to provide various intelligent services to the users. Most of IoT services, such as air conditioning and lighting management, detect the current states of things and environments through sensors and make reasonable decisions based on the data for the specific situations. As the number and functions of the things increase, the relationship between the things has become more complicated. This complexity makes it difficult to monitor and manage the data of things precisely, which leads to difficulties in proper control of things.

Cyber-physical systems (CPS), or *digital twin*, may be a solution for the problem [1]. The digital twin is considered an element of CPS because it means the mirror image of a physical thing. In this paper, we use the term *cyber thing* as the digital twin. CPS has some aspects of IoT in terms of using data of physical things collected through the Internet. The control devices, mechanical body, and corresponding

cyber things are connected to fully operate the corresponding physical systems [2]. By virtue of the tight interworking between cyber things and physical things, CPS can be used for many critical applications, including smart city, smart factory, and smart grid, which require accurate analysis and control functionalities based on physical systems and the environments [3–5]. Since cyber things inherit the exact features and states of the physical things, they can be used for accurate estimation, prediction, and control of the physical things by means of dynamics simulations and the other computational means.

The emergence of *virtual reality* (VR) technologies makes it possible to provide services by immersion into computer-generating virtual environments [6]. Currently, most of the VR services are provided in the virtual environment that has no direct relations with the physical environment. Although *augmented reality* (AR) has more information of the real world than VR, it just overlays cyber things or cyber information on the physical things or the physical space. If the cyber thing or space is actually connected to the physical thing

or space, users in the cyber space can vividly experience more immersive VR/AR services. Therefore, the integration of CPS and VR/AR empowers the legacy VR/AR services to dramatically enhance the reality and the interactivity with real world [7]. In this paper, we propose a new type of education service called virtual reality-based cyber-physical education system (VR-CPES) to support the VR/AR education services based on CPS.

1.1. Scenario. We introduce a new driver-training service scenario in order to explain, in detail, what VR-CPES can provide for efficient training in wireless/mobile environments. The VR-CPES services are executed on a mobile device, which is a sort of the VR device. A comparison between the advantages of the services implemented in VR-CPES and the existing services execution environments is shown in Table 1.

Figure 1 shows the scenario and architecture of VR-CPES driver-training service. VR-CPES driver training is very safe from risks, such as vehicle accidents. In addition, the constraints of the training caused by the lack of equipment and space are eliminated by virtue of the mobile platform. Subsequently, users can practice in a variety of driving environments, such as where they wanted to practice and where the steering-wheel position changes according to the country. On the physical side, using actual data allows user to practice in a realistic environment because conditions of the vehicle are immediately reflected by temperature, weather, and time changes in the virtual environment. The services listed above require not only generation of the virtual environment, but also interworking of cyber-physical environments. Therefore, the VR-CPES mobile platform needs to satisfy the following requirements.

1.2. Requirements. In this section, we suggest the following requirements to solve the problems shown in the scenario. In VR/AR education services, physical things must be represented in cyber space as cyber things, and the two assets must be interworking. As mentioned above, the cyber things are called the digital twin. A digital twin is a digital replica of a physical thing that conducts all of its functionalities [8]. For example, vehicles, traffic signs, and even people in physical space can be reflected by the digital twin in cyber space. In the general IoT service, the physical things usually gather environment data and use them in mobile devices. However, the physical things in VR-CPES use digital twin models so that monitoring status and management are improved compared with physical things in general IoT [9, 10].

Because VR-CPES uses a digital twin in cyber space, it needs to generate and manage various digital twin models in cyber space, according to various physical spaces. An IoT platform considering a digital twin needs to manage resources of things based on time because physical things should be reflected by the digital twin in real time. A general IoT platform does not sufficiently support such time-based resource management. Moreover, the IoT platform for VR-CPES must consider cloud and edge computing, which is

responsible for generating, simulating, and analyzing the digital twin because a single mobile device cannot manage the uncountable resources of the physical thing and digital twin, respectively. In other words, the *quality of experience* (QoE) of a user in VR-CPES is totally related to the interworking of the physical thing and digital twin.

Reliability of VR-CPES increases as the virtual environment and digital twin become more reliable. The reliability of the virtual environment and digital twin is closely related to the seamless interworking of cyber-physical environments. It also depends on a dependable transmission of physical space and physical-thing data. Table 2 shows the network *quality of service* (QoS) requirements for data which is transmitted in VR-CPES. The VR-CPES needs to satisfy QoS requirements for all applications in the cyber-physical environments. Since the main things in our scenario are vehicles and VR devices, we investigated the traffic class based on this [11–13].

In this paper, we propose a novel VR-CPES platform to provide virtual reality contents for efficient education in the mobile environment, as mentioned in the above scenario, and a network framework that can satisfy the requirements for all contents. The remainder of the paper is organized as follows: in Section 2, we summarize IoT platform, SDN, and TSN and describe insufficiency of each technology for the VR-CPES service; in Section 3, we propose the VR-CPES platform by describing each component of the platform and the real-time network framework for seamless VR-CPES service; Section 4 verifies the performance of the proposed real-time network framework; finally, Section 5 concludes this paper.

2. Related Works

2.1. IoT Platform. In order to apply IoT technologies to various domains, such as smart city and health care, IoT platforms are being developed that can adaptively operate in environments where many things need to be monitored and managed [14–16]. In standard organizations, such as oneM2M and OCF design, IoT has standards for the standardized operation of platforms, and these standard-based platforms facilitate the connection of different IoT devices and resource management [17, 18]. Various commercial platforms, such as GE's Predix platform and IBM's Watson IoT platform, are developed as platforms considering the digital twin [19, 20]. They support IoT services for the digital twin using Predix Machine, Asset, and Machine Data Analytics. In addition, multiple technologies have incorporated an IoT platform to replicate physical assets of a digital twin and utilize them [2]. Due to the characteristic of the digital twin that is a digital replica of physical assets, the IoT platforms for the digital twin must consider communication network for seamless data transmission of cyber and physical things.

2.2. Software-Defined Networking. As mentioned above, reliability of VR-CPES is based on dependable data transmission, according to the dynamically changing interconnection of cyber-physical things. *Software-defined*

TABLE 1: Comparison of driver-training services.

Conditions	Real training	VR simulator	VR-CPES driver training
Safety under training	Low	High	High
Volume of training space	Large	Small	Small
Diversity of training scenario	Low	Middle	High
Reality of training	High	Low	High

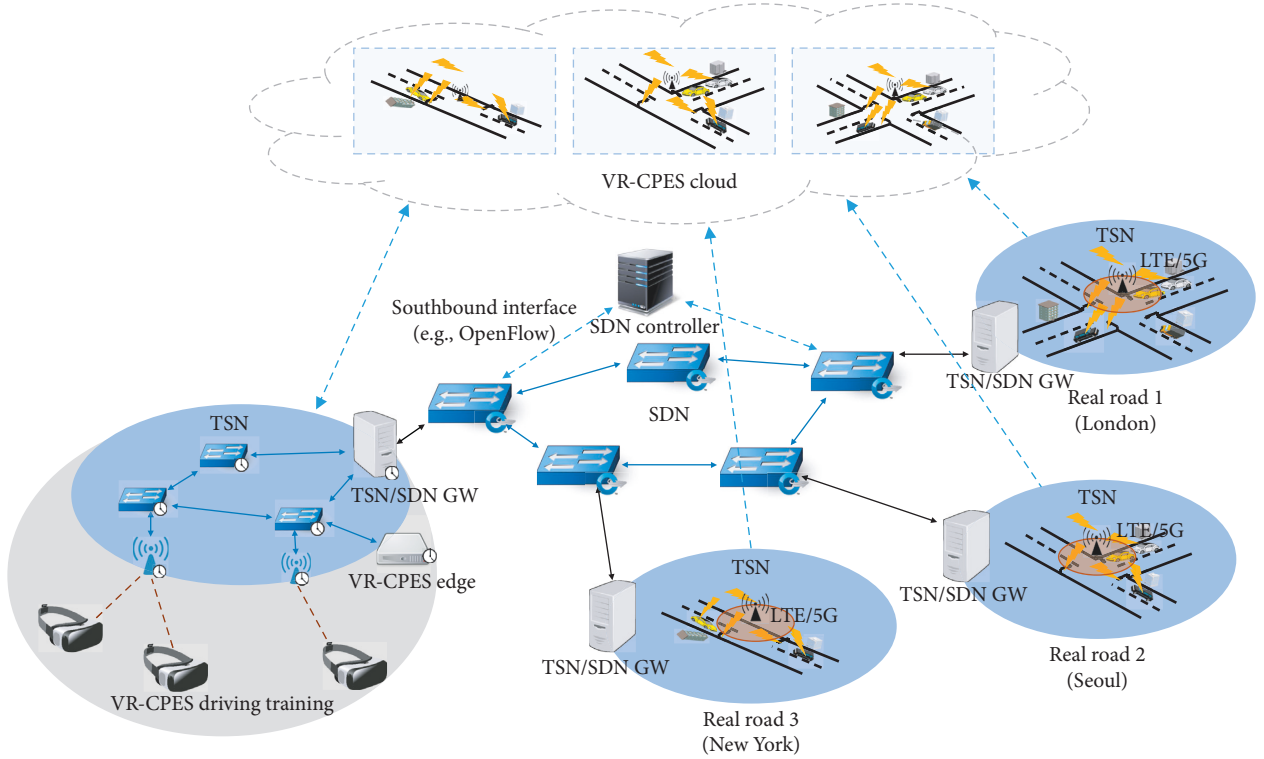


FIGURE 1: VR-CPES driver-training service.

networking (SDN) is a centralized network technology with a global network view that allows quick and dynamic configuration of the network, depending on the complex quality of service (QoS) of various network applications [21]. The SDN separates the control plane, which determines the network policy, from the data plane, which forwards the actual data. This decoupling of planes abstracts low-level network functionality and higher level service, which facilitates programmable network configuration. This feature is suitable for a large-scale IoT, which, for a network to be configured adaptively, depends on the different network requirements of many things [22]. In the SDN, the control plane receives network information of the data plane from a southbound interface, and the OpenFlow protocol is used dominantly for this operation [23]. The OpenFlow switch can identify all of the layer 2, layer 3, and layer 4 (L2–L4) protocols when configuring the path (i.e., the flow), so the SDN has the advantage of flow configuration in the reconfigurable and heterogeneous network condition [24]. However, it is difficult to calculate proper data transmission paths in systems where control data need to be transmitted with critical network requirements, such as CPS. Various studies have been conducted to extend SDN suitable for these systems [25]. In the

network for VR-CPES, it is necessary to transfer data depending on the QoS requirements of different applications, which are simulation for digital twin and VR rendering. In addition, time-based data transmission should be considered for seamless operation of a digital twin and physical thing.

2.3. Time-Sensitive Networking. Time-sensitive networking (TSN) is a set of standards under development by the Time-Sensitive Networking Task Group which is part of the IEEE 802.1 Working Group [26]. In the TSN standard, various technologies, such as time synchronization and frame preemption, are being developed to provide deterministic network service of low latency and low packet loss in ethernet. The predecessor of the TSN standard technology is the audio-video bridging (AVB), which started in the media industry to transmit high-quality audio and video data in real time using a standardized-ethernet network. Subsequently, there has been a requirement to transfer control/management data through the ethernet network in industry domains, such as automotive and factory automation. The standard is being extended to develop a reliable network based on time synchronization for each industry's

TABLE 2: QoS requirements for applications in VR-CPES.

Traffic class	Bandwidth	End-to-end delay	Jitter
Control data (vehicle)	Low	2.5 ms	Sub-microsecond
Safety data (vehicle)	Medium	45 ms	Sub-microsecond
Infotainment data (vehicle)	High	150 ms	100 ms
Video data (VR)	80 k~50 Mbps	<100~500 ms	<50~150 ms
Audio data (VR)	10~80 kbps	<100 ms	<100 ms

domain network. With the motivations developed to cover the requirements of various industries, TSN is utilized as a network technology in a system that requires transmission of critical control data and high-bandwidth data [27]. In addition, deterministic network technologies of TSN are being applied in the mobile network environment. In particular, the development of a standard for a fronthaul and various studies for mobile networks is in progress to meet the strict reliable low latency communication requirement of 5G network [28]. However, until now, TSN standards have a limitation of local area network due to problems such as time synchronization accuracy. In order to solve it, the IETF is in the process of extending the TSN to the wide-area network under the name DetNet [29], and the VR-CPES service also requires a dynamic and large-scale network environment, so extended technologies of the existing TSN are needed.

3. The Proposed VR-CPES Platform

3.1. VR-CPES Platform. The proposed VR-CPES platform is shown in Figure 2. The VR-CPES platform consists of four components: VR device, VR-CPES edge, VR-CPES cloud, and physical things. Each component transmits data through a TSN/SDN real-time network framework, which is described in Section 3.2. For the ease of explanation, we describe the operation of each component of the VR-CPES platform based on the driver-training service scenario in the introduction, as shown in Figure 3.

3.1.1. VR Device. The VR-CPES service should operate by interworking the corresponding vehicle according to the user's operation and the digital twins, which are reflected in the virtual environment from physical space. For this reason, in the VR device, it is very difficult to calculate the results for rendering—such as the perspective view associated with each digital twin according to the user's action—in order to immerse the user in the virtual environment. Therefore, an application that requires high-computing resources, such as a simulation, is operated in the VR-CPES edge and cloud. The VR device transmits only the user's behavior and eye tracking as sensor data and receives external-calculated data of digital twins and the virtual environment and then renders it.

3.1.2. VR-CPES Edge. The VR-CPES edge is a subsimulator for each VR device (i.e., each user). It estimates the commands of the user for the user's digital twin vehicle (DTV), based on the data received from the VR device. Since the

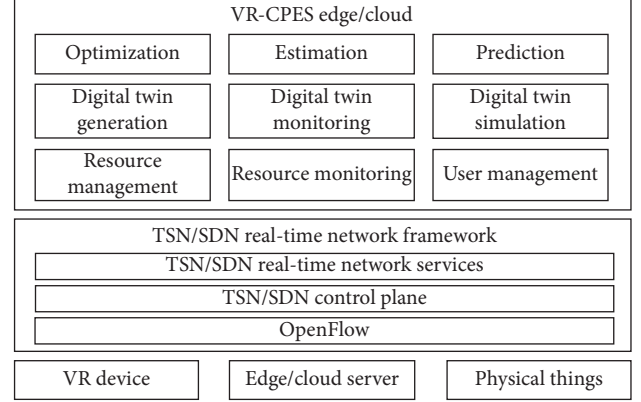


FIGURE 2: VR-CPES platform.

VR-CPES training service is intended for the person, it should operate at less than 13 ms, which does not affect human perception [30]. In other words, the control of the corresponding DTV interworking with the movement of the user, which should be executed in real time for the VR experience, needs to be simulated on the VR-CPES edge rather than the cloud. The VR-CPES edge performs simulation by receiving external environment data and previously executed simulation data for the scenario selected by the user in the VR-CPES cloud. The result data of simulation are transmitted back to the VR-CPES cloud and updated for analysis of DTV.

3.1.3. VR-CPES Cloud. The VR-CPES cloud is the main simulator of the VR-CPES platform and performs functions such as management and monitoring of VR-CPES things. In the cloud simulator, virtual environment models and digital twin models are generated based on data collected from physical space. In the VR-CPES cloud, unlike the VR-CPES edge, which considers only the user's DTV, the environment model and other digital twin models must interact with each other in the simulator. Therefore, simulation should be performed in a distributed cosimulation environment, rather than a single simulation environment [31]. Digital twin models are designed and interpreted based on the large amount of physical-space data, so that they can not only optimize the states of the corresponding things, but estimate and predict certain situations for appropriate control. This series of processes requires various analytical techniques, such as stochastic approach, machine learning, AI, and appropriate control theory. The VR-CPES cloud stores these simulated results of each situation, and the data are used as

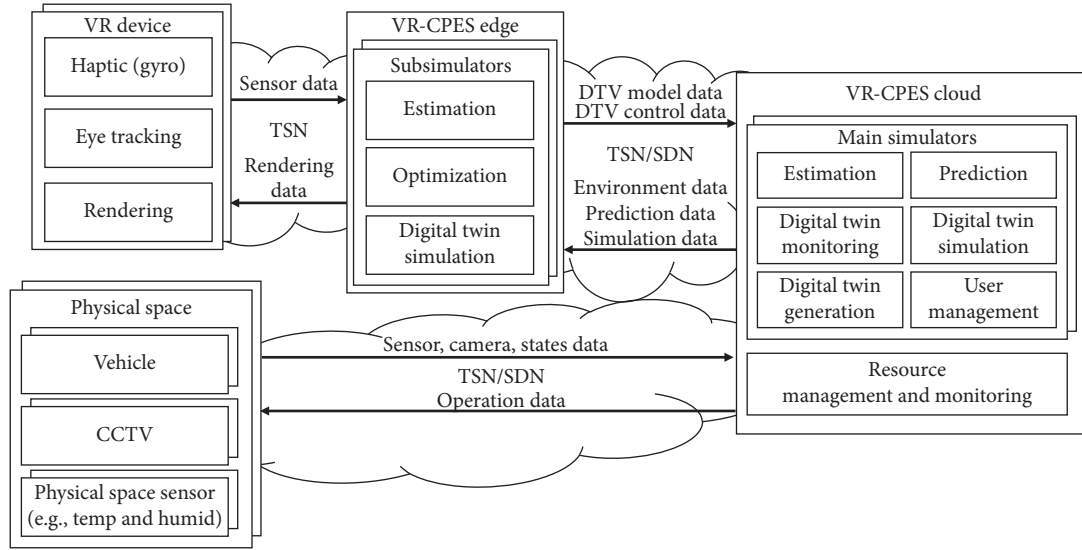


FIGURE 3: Structure of VR-CPES platform.

external environment input data of the DTV model in VR-CPES edge's subsimulator.

3.1.4. Physical Space. Physical space is a real physical space interworked with virtual reality environments. In a scenario, they are real road environments such as New York, Seoul, and London. Data of physical things that occur in the physical space are transmitted to the VR-CPES cloud and used to generate virtual space. Each physical thing must be utilized as digital twins, not only in the monitoring, but in the simulator. Therefore, unlike legacy IoT platform, real-time data transmission must be considered in VR-CPES platform, and this is supported by TSN/SDN real-time network framework.

3.2. TSN/SDN Real-Time Network Framework (TSRtnet). As mentioned above, the service of the VR-CPES platform requires real-time data transmission for VR experience of users. It can be achieved, not only by resource management according to the QoS of application level in each platform component, but by supporting data transmission according to the QoS of network level. TSN, a standard for real-time data transmission in ethernet networks, transmits data in frame units based on priority. High-priority frames are transmitted in further advance than low-priority frames through a TSN switch, and there are several algorithms to determine how ethernet frames are processed and how priority is assigned. The frames, which require hard real-time data transmission, such as physical states data in the VR-CPES service, are processed based on a time-aware shaper algorithm in TSN [32]. This time-based, scheduled-traffic transmission requires time synchronization between the nodes, which configure the network. The Precision Time Protocol (PTP) is used for time synchronization between nodes under sub-microsecond accuracy by means of hardware timestamps and compensating the time offset considering link delay [33]. In addition, the TSN

standards are extended for requirement of time-sensitive applications such as a multiple time domain [34].

The main application for the VR-CPES service runs on the VR-CPES cloud, not on the VR device, which is physically separated from the VR device runtime environment. It means the local network needs to extend to a large-scale network for the user's experience. TSN can guarantee time-sensitive data transmission, but it is limited to less than 7 hops switched network due to the inaccuracy of time synchronization. Although the TSN standard has recently been applied to the factory network considering the connection between TSN nodes based on the centralized network configuration concept in the industrial network, it is still not enough to satisfy the requirements of VR-CPES service [35]. Because it is difficult to practically configure the entire network with homogeneous TSN, we suggest TSN/SDN real-time network framework to plug multiple TSNs into SDN core network.

The structure of TSN/SDN real-time network framework (TSRtnet) for the VR-CPES platform is shown in Figure 4. The architecture consists of TSN/SDN gateways, an SDN controller, and OpenFlow switches. Each TSN is connected to an external SDN through the TSN/SDN gateway, which is managed by the SDN controller.

3.2.1. TSN/SDN Gateway. The TSN/SDN gateway is responsible for connecting the local TSN to the SDN backbone network. In this paper, we assume that the TSN/SDN gateway is capable of processing OpenFlow protocol to communicate with the SDN controller. The TSN nodes register and reserve time-sensitive streams for communication as talker and listener using Stream-Reservation Protocol (SRP). The SRP parameters used for TSN stream management requiring guaranteed QoS are shown in Table 3 [36]. The TSN nodes, such as physical things and the VR device, which need to be communicated with the VR-CPES cloud through SDN, reserve the TSN streams to the

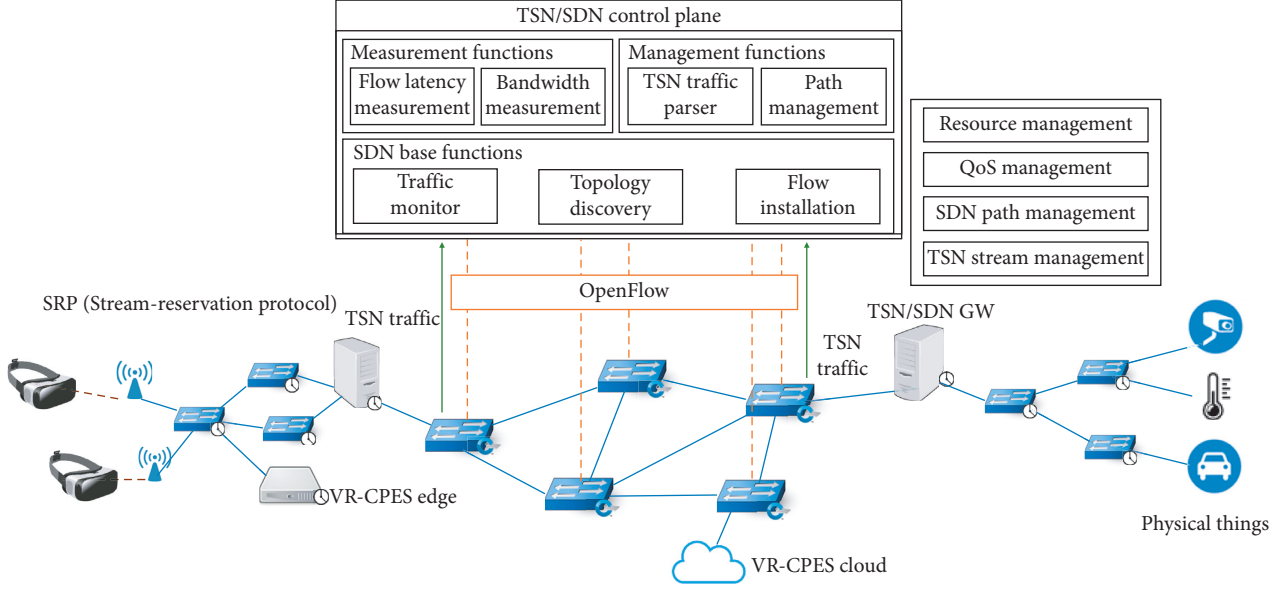


FIGURE 4: TSN/SDN real-time network framework.

TABLE 3: Stream-Reservation Protocol parameters.

StreamID	Stream identifier
Stream destination address	Destination address
Stream VLAN ID	VLAN identifier
MaxFrameSize	Traffic specification associated with a stream
MaxIntervalFrames	Traffic specification associated with a stream
Data frame priority	Priority value
AccumulatedLatency	Worst-case latency that a stream can encounter

TSN/SDN gateway by SRP. The TSN/SDN gateway encapsulates the SRP message of the connected node into a packet destined for the VR-CPES cloud. This packet is sent as a packet-in message to the SDN controller, and the SDN controller sets the flow of the SDN by parsing the SRP contained in this message.

3.2.2. SDN Controller. The SDN controller parses the packet-in message received from the TSN/SDN gateway and configures a path to communicate with VR-CPES cloud. The SDN controller has three base functions: traffic monitor, topology discovery, and flow installation. These functions set flows for configuring the path [37]. As the name implies, each function is responsible for monitoring traffic of the data plane, discovering network topology, and installing flow for packet forwarding of the data plane, respectively. Additionally, we designed four SDN functions to forward time-sensitive traffic to SDN core network:

- (1) *Flow latency measurement.* This function measures the flow latency of each link to establish a network path according to a requirement of the application's end-to-end delay. The SDN controller has a global view of each flow latency through the

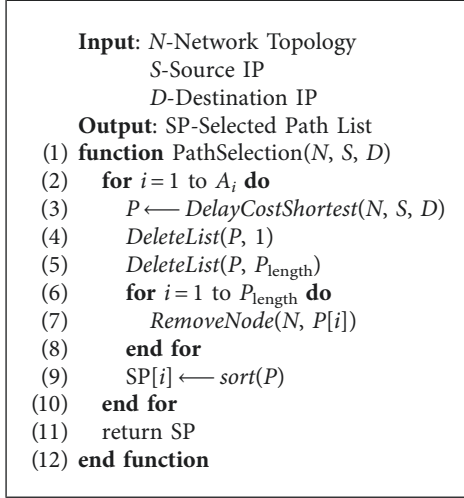
latency-measurement function [38]. In this function, the SDN controller measures the latency using Internet Control Message Protocol (ICMP) every second.

- (2) *Bandwidth measurement.* This function uses the bandwidth-measurement function to measure available bandwidth (ABW) of i -th link in SDN. Available bandwidth is calculated by the amount of packets forwarding to SDN flow in specific time. The amount of forwarded packets is sent to the controller through FlowStates message, and then the controller calculates available bandwidth using the period time T when the controller receives FlowStates message:

$$b_i(t) = \frac{\text{count}_i(t) - \text{count}_i(t - T)}{T}, \quad (1)$$

$$\text{ABW}_i = c_i(1 - b_i).$$

- (3) *TSN traffic parser.* This function is parsing the time-sensitive traffic, which is received from TSN/SDN gateway by OpenFlow packet-in message. The SDN controller parses the QoS of applications by checking the SRP parameters and VLAN header. These data are used as QoS information of applications to set a path in a path management function.
- (4) *Path management.* Path management function sets path from source to destination based on the calculated latency and bandwidth of flows by measurement functions. Algorithm 1 shows the TSN/SDN path selection algorithm for selecting the candidate path from source to destination. Subsequently, the final path is selected from among the selected paths from Algorithm 1 using a path configuration flowchart. The TSN/SDN path configuration flowchart is shown in Figure 5.



ALGORITHM 1: Pseudocode of the TSN/SDN path selection algorithm.

3.2.3. TSN/SDN Path Selection Algorithm. The path selection algorithm we propose for TSRTnet is shown in Algorithm 1. In this algorithm, N is the network topology, S is the source IP, and D is the destination IP. The data-stream list to be connected according to application's QoS requirement is represented by A . Again, there are three types of data stream: physical states data, audio data, and video data. The DelayCostShortest function calculates the shortest path from S to D , and N assigns the cost to the link latency [39]. The selected path list as output of algorithm is SP.

3.2.4. TSN/SDN Path Configuration Flowchart. The SDN controller configures the data-stream path as flow in a flow table of SDN core network, using the selected path (SP) list, which is the output of path selection algorithm. DS is a data-stream list, which is the same as A in the path selection algorithm, and it contains the application's QoS requirement. In the VR-CPES service, the TSN data streams are transmitted based on priority, which are identified by VLAN. In the TSN/SDN path configuration, the TSN data stream with highest priority is configured in advance. The SP and DS are ordered according to the highest-priority streams.

3.2.5. OpenFlow Switch Operation by SDN Controller. The SDN controller processes the time-sensitive traffic and management path using OpenFlow protocol. The SDN controller sets the path based on the packet received in the OpenFlow packet-in message, and the path is divided into VLANs. As mentioned above, the QoS requirement of each application is classified by the SRP and VLAN of the TSN in the TSN traffic parser function. We assume that the SDN controller knows the QoS settings of each TSN for path configuration. The SDN controller sets the path, which is chosen from the TSN/SDN path configuration function, as flow using VLAN in the flow table of each OpenFlow switch. If the QoS of the application cannot be satisfied in the

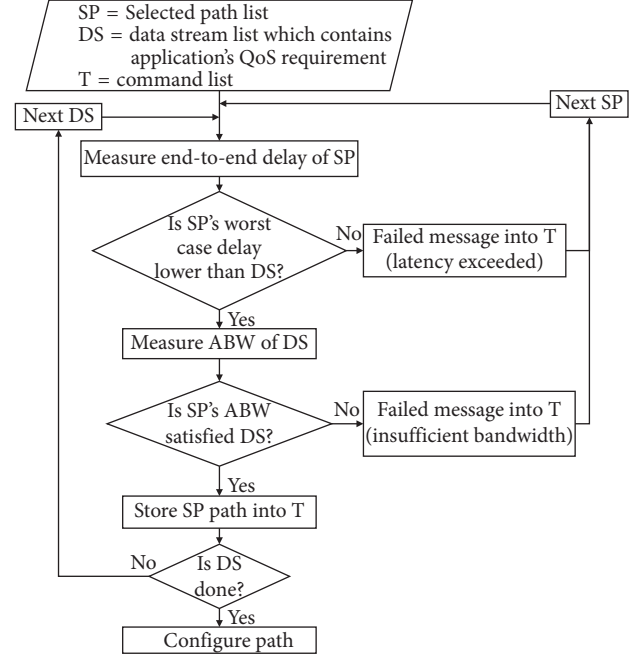


FIGURE 5: TSN/SDN path configuration.

network, the SDN controller sends the corresponding fail message to the TSN/SDN gateway.

4. Experiments and Evaluation

In VR-CPES driver-training service, physical states data, audio data, and video data require transmitting in real time, and all QoS requirements in network are mentioned above. To verify the real-time data transmission on the VR-CPES platform, we compared TSRTnet traffic with best-effort traffic. The experiment is performed under the Mininet emulator [40]. Figure 6 shows the network topology of the experiment. In the topology, each link bandwidth is 10 Mbps, and latency is 3 ms. Host 1 (H1) and Host 2 (H2) are the sources, and Host 3 (H3) is the receiver. We consider sources as the TSN/SDN gateways, and the receiver as the VR-CPES cloud of the VR-CPES driver-training service. Both H1 and H2 transmit 6 Mbps TSN streams to H3, which consists of 0.8 Mbps physical states data, 0.2 Mbps audio data, and 5 Mbps video data, respectively.

Figure 7 shows the packet loss of TSRTnet traffic and best-effort traffic. A packet loss that we measured is the sum of the loss packets in all three types of data delivered from the sources H1 and H2 to the receiver H3. In the proposed TSRTnet, the packet loss occurred almost 0% in both H1 (0.4%) and H2 (0.5%), but in the best-effort traffic case, a packet loss of 7% in H1 and 18% in H2 occurred. In the method of TSRTnet, packet loss scarcely occurred in all three data streams because the path is set by comparing the priority and reserving the bandwidth. On the contrary, in the case of best effort, all data are delivered with the same condition, which means they do not consider the QoS requirement. Therefore, the data with the highest bandwidth, such as video data, cause overhead on a certain link, and it incurs a large

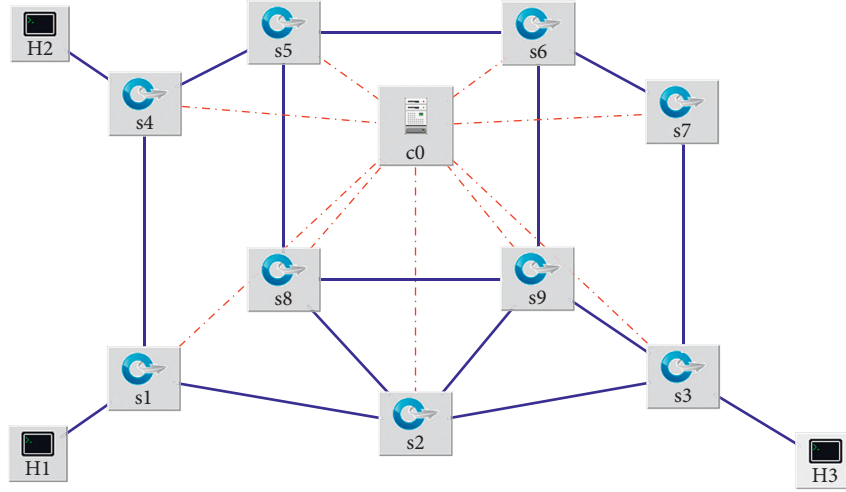


FIGURE 6: Network topology.

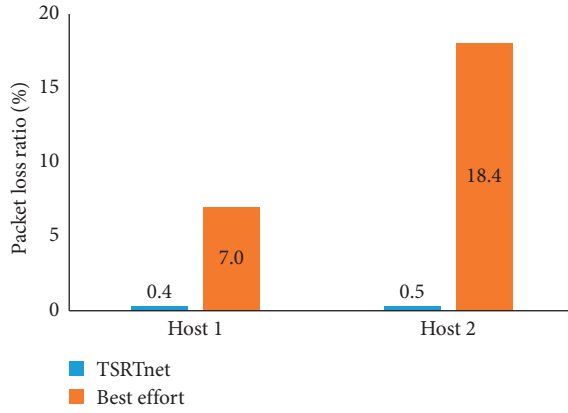


FIGURE 7: Packet loss of TSRTnet traffic and best-effort traffic.

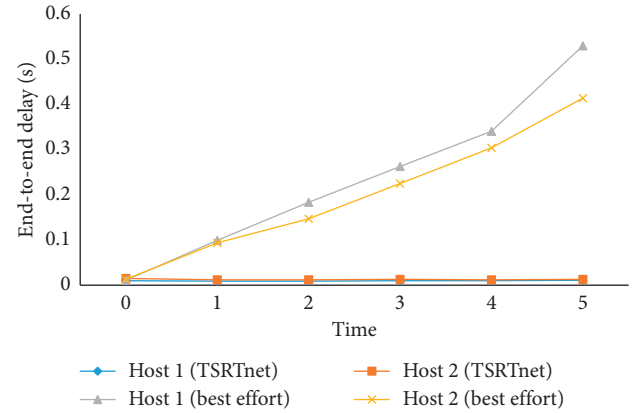


FIGURE 8: End-to-end delay of TSRTnet traffic and best-effort traffic.

amount of packet loss. The reason why the packet loss of H1 is lower than H2 is because the hop count of H1 is less than H2.

Figure 8 shows the comparison end-to-end delay between TSRTnet and best-effort traffic. In particular, we measured physical states traffic because it is highest priority traffic in our experiment. Both end-to-end delay variations of physical states traffic from H1 and H2 are constant in TSRTnet. Except for the delay due to the initial path configuration, it represents that the delay variation is less than 2 ms. On the contrary, the end-to-end delay of best-effort traffic increases continuously compared with the results from TSRTnet traffic. Until 0.2 seconds before packet loss, the end-to-end delay of best-effort was stable, as with TSRTnet traffic. Since then, however, the best-effort traffic delay for the three types of streams exceeded 100 ms in 1.1 seconds, which means it does not meet the requirements of the video stream with the highest end-to-end delay requirements. In addition, since H1 is close to the receiver, the traffic of H2 caused congestion in the same paths of H1. Consequently, the end-to-end delay of H1 is increased faster than H2.

The results show that the TSRTnet framework is more suitable for the VR-CPES service than the conventional

network. As mentioned above, for seamless VR-CPES service, physical-state data, audio data, and video data are transmitted together, which means that data transmission according to complex QoS should be guaranteed in the network. TSRTnet shows almost zero packet loss compared to the conventional network, which transmits data with best effort. Packet loss is closely related to VR service interruption, which has a fatal impact on the user's VR experience. In addition, the constant latency variation of TSRTnet can satisfy the transmission of physical states data for time-critical simulation in VR-CPES.

5. Conclusion

In this paper, we proposed a VR-CPES platform that can provide novel VR-based educational services. VR-CPES platform solves the QoE problem of users caused by delay and loss of data transmission based on TSN/SDN real-time network, which makes time-critical data transmission between cyber things and physical things. Since VR-CPES services require strict time requirements, as well as bandwidth of multimedia and states data of physical things, the

time-sensitive network services over the entire network, including core networks and local networks, are essential. The proposed TSN/SDN real-time network framework, called TSRTnet, effectively supports the requirements in terms of packet-loss ratio and end-to-end delay. TSRTnet includes a gateway function for the interworking between SDN core networks and TSN local networks, as well as a path selection algorithm for selecting candidate paths. As a result, TSRTnet showed stable performance of end-to-end delay variation within 2 ms and fewer packet losses compared to conventional best-effort network, which experienced very large delays and higher rates of packet loss.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper is partially supported by the Institute for Information and Communications Technology Promotion (IITP) funded by the Ministry of Science and ICT (MSIT) (2015-0-00816-004) and by the National Research Foundation of Korea (NRF) grant (no. 2017010875) from the Korea government.

References

- [1] E. A. Lee, "CPS foundations," in *Proceedings of 47th ACM/IEEE Design Automation Conference on-DAC'10*, pp. 737–742, Anaheim, CA, USA, June 2010.
- [2] M. Schluse, M. Priggemeyer, L. Atorf, and J. Rossmann, "Experimentable digital twins—streamlining simulation-based systems engineering for industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 6, pp. 1722–1731, 2018.
- [3] J. Lee, B. Bagheri, and H. A. Kao, "A cyber-physical systems architecture for industry 4.0-based manufacturing systems," *Manufacturing Letters*, vol. 3, pp. 18–23, 2015.
- [4] F. Salim and U. Haque, "Urban computing in the wild: a survey on large scale participation and citizen engagement with ubiquitous computing, cyber physical systems, and internet of things," *International Journal of Human-Computer Studies*, vol. 81, pp. 31–48, 2015.
- [5] X. Yu and Y. Xue, "Smart grids: a cyber-physical systems perspective," *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1058–1070, 2016.
- [6] N. Gavish, T. Gutiérrez, S. Webel et al., "Evaluating virtual reality and augmented reality training for industrial maintenance and assembly tasks," *Interactive Learning Environments*, vol. 23, no. 6, pp. 778–798, 2015.
- [7] U. H. Govindarajan, A. J. C. Trappey, and C. V. Trappey, "Immersive technology for human-centric cyberphysical systems in complex manufacturing processes: a comprehensive overview of the global patent profile using collective intelligence," *Complexity*, vol. 2018, Article ID 4283634, 17 pages, 2018.
- [8] T. H. J. Uhlemann, C. Lehmann, and R. Steinhilper, "The digital twin: realizing the cyber-physical production system for industry 4.0," *Procedia CIRP*, vol. 61, pp. 335–340, 2017.
- [9] S. Weyer, T. Meyer, M. Ohmer, D. Gorecky, and D. Zühlke, "Future modeling and simulation of CPS-based factories: an example from the automotive industry," *IFAC-PapersOnLine*, vol. 49, no. 31, pp. 97–102, 2016.
- [10] K. M. Alam and A. E. Saddik, "C2PS: A digital twin architecture reference model for the cloud-based cyber-physical systems," *IEEE Access*, vol. 5, pp. 2050–2062, 2017.
- [11] L. Skorin-Kapov, D. Huljenic, D. Mikic, and D. Vilendecic, "Analysis of end-to-end QoS for networked virtual reality services in UMTS," *IEEE Communications Magazine*, vol. 42, no. 4, pp. 49–55, 2004.
- [12] S. Tuohy, M. Glavin, C. Hughes, E. Jones, M. Trivedi, and L. Kilmartin, "Intra-vehicle networks: a review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 534–545, 2015.
- [13] Y. Chen, T. Farley, and N. Ye, "QoS requirements of network applications on the Internet," *Information Knowledge Systems Management*, vol. 4, no. 1, pp. 55–76, 2004.
- [14] G. Yang, L. Xie, M. Mäntysalo et al., "A health-IoT platform based on the integration of intelligent packaging, unobtrusive bio-sensor, and intelligent medicine box," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2180–2191, 2014.
- [15] R. Lea and M. Blackstock, "City hub: a cloud-based IoT platform for smart cities," in *Proceedings of 2014 IEEE 6th International Conference on Cloud Computing Technology and Science (CloudCom)*, pp. 799–804, Hong Kong, China, December 2014.
- [16] N. H. Motlagh, M. Bagaa, and T. Taleb, "UAV-based IoT platform: a crowd surveillance use case," *IEEE Communications Magazine*, vol. 55, pp. 128–134, 2017.
- [17] oneM2M Standardisation Committee, *oneM2M Release 2 Specifications*, oneM2M, 2016, <http://www.onem2m.org/technical/published-drafts>.
- [18] OIC Specification 1.1, 2016, <https://openconnectivity.org/resources/specifications>.
- [19] GE Predix IoT Platform, 2017, <https://www.ge.com/digital/predix>.
- [20] C. Mercer, *Internet of Things Platforms: Azure, AWS, IBM Watson and More-Which is the Best IoT Platform for Your Business*, Computer World, 2016, <https://www.computerworlduk.com/galleries/data/best-internet-of-things-platforms-3635185/>.
- [21] S. Bera, S. Misra, and A. V. Vasilakos, "Software-defined networking for Internet of things: a survey," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1994–2008, 2017.
- [22] A. Hakiri, P. Berthou, and A. Gokhale, "Publish/subscribe-enabled software defined networking for efficient and scalable IoT communications," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 48–54, 2015.
- [23] ONF, *OpenFlow Switch Specific Version 1.5.1*, Open Networking Foundation, 2015, <https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf>.
- [24] S. Paris, A. Destounis, L. Maggi, G. S. Paschos, and J. Leguay, "Controlling flow reconfigurations in SDN," in *Proceedings of IEEE INFOCOM 2016-the 35th Annual IEEE International Conference on Computer Communications*, pp. 1–9, San Francisco, CA, USA, April 2016.
- [25] R. W. Skowrya, A. Lapets, A. Bestavros, and A. Kfoury, "Verifiably-safe software-defined networks for CPS," in *Proceedings of the 2nd ACM International Conference on High*

- Confidence Networked Systems*, pp. 101–110, Philadelphia, PA, USA, April 2013.
- [26] Time-Sensitive Networking Task Group, 2018, <http://www.ieee802.org/1/pages/tsn.html>.
 - [27] M. Wollschlaeger, T. Sauter, and J. Jasperneite, “The future of industrial communication: automation networks in the era of the Internet of things and industry 4.0,” *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 17–27, 2017.
 - [28] D. Chitimalla, K. Kondepudi, L. Valcarenghi, M. Tornatore, and B. Mukherjee, “5G fronthaul-latency and jitter studies of CPRI over ethernet,” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 9, no. 2, pp. 172–182, 2017.
 - [29] Deterministic Networking, 2018, <https://datatracker.ietf.org/wg/detnet/charter/>.
 - [30] E. Bastug, M. Bennis, M. Médard, and M. Debbah, “Toward interconnected virtual reality: opportunities, challenges, and enablers,” *IEEE Communications Magazine*, vol. 55, no. 6, pp. 110–117, 2017.
 - [31] S. Yun, J. H. Park, and W. Kim, “Data-centric middleware based digital twin platform for dependable cyber-physical systems,” in *Proceedings of 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, Milan, Italy, July 2017.
 - [32] IEEE Std. 802.1Qbv, *Standard for Local and Metropolitan Area Networks-Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks Amendment: Enhancements for Scheduled Traffic*, IEEE, Piscataway, NJ, USA, 2016.
 - [33] IEEE Std. 1588-2008, *Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, IEEE, Piscataway, NJ, USA, 2008.
 - [34] IEEE Std. 802.1AS, *IEEE Standard for Local and Metropolitan Area Networks-Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks*, IEEE, Piscataway, NJ, USA, 2010.
 - [35] IEEE Std. 802.1Qcc, *IEEE Approved Draft Standard for Local and metropolitan Area Networks-Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks Amendment: Stream Reservation Protocol (SRP) Enhancements and Performance Improvements*, IEEE, Piscataway, NJ, USA, 2018.
 - [36] IEEE Std. 802.1Qat, *IEEE Standard for Local and Metropolitan Area Networks, Virtual Bridged Local Area Networks, Amendment 14: Stream Reservation Protocol*, IEEE, Piscataway, NJ, USA, 2010.
 - [37] H. S. Kim, S. Yun, H. Kim, H. Shin, and W. Kim, “An efficient SDN multicast architecture for dynamic industrial IoT environments,” *Mobile Information Systems*, vol. 2018, Article ID 8482467, 11 pages, 2018.
 - [38] D. Sinha, K. Haribabu, and S. Balasubramaniam, “Real-time monitoring of network latency in software defined networks,” in *Proceedings of 2015 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pp. 1–3, Kolkata, India, December 2015.
 - [39] F. Kuipers, T. Korkmaz, M. Krunz, and P. Van Mieghem, “Performance evaluation of constraint-based path selection algorithms,” *IEEE Network*, vol. 18, no. 5, pp. 16–23, 2004.
 - [40] R. L. S. de Oliveira, A. A. Shinoda, C. M. Schweitzer, and L. R. Prete, “Using Mininet for emulation and prototyping software-defined networks,” in *Proceedings of the 2014 IEEE Colombian Conference on Communications and Computing (COLCOM)*, pp. 1–6, Bogota, Colombia, June 2014.

