Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8829868, 8 pages
https://doi.org/10.1155/2020/8829868

Research Article

LSEA: Software-Defined Networking-Based QoS-Aware Routing Mechanism for Live-Soccer Event Applications in Smart Cities

Yingcheng Zhang[1] and Gang Zhao[2]

[1]*Northeast Normal University, Changchun 130024, China*
[2]*Shenzhen University, Shenzhen 518060, China*

Correspondence should be addressed to Gang Zhao; spjava111@163.com

Received 26 August 2020; Revised 6 October 2020; Accepted 11 October 2020; Published 7 November 2020

Academic Editor: Jianhui Lv

The smart cities provide a better connection between services and citizens based on new Internet technologies. During the building process of smart cities, some burgeoning applications have been emerging and changing the daily lifestyle of people, e.g., live streaming applications. Especially, the live-soccer event applications have attracted much attention and can improve people's enjoyment of life to a great extent, such as the Europe five major league matches and FIFA world cup. For such applications, the traditional routing strategies cannot do Quality-of-Service (QoS) awareness, and thus, the network performance and the Quality of Experience (QoE) of users cannot be guaranteed. In this paper, we employ Software-Defined Networking (SDN) to make QoS awareness for the special live-soccer event applications, in which the QoS-aware routing mechanism is proposed, called LSEA. Meanwhile, delay, delay jitter, and packet loss rate are considered as three objects. On this basis, the improved Dijkstra routing algorithm and SDN-based disjoint routing algorithm are devised. Finally, the proposed LSEA is implemented over Mininet, and the experimental results demonstrate its feasibility and efficiency.

1. Introduction

With the rapid development of new Internet technologies such as mobile edge computing [1], 5G [2], and Internet of Things (IoT) [3], the building of smart cities [4, 5] has attracted much attention from many countries and regions, which can provide a better connection between services and citizens based on the combination of these new Internet technologies. At the same time, some burgeoning applications have been emerged and changed people's daily life patterns. For example, there have been a lot of live-broadcasting platforms [6], such as TikTok, Livestream, Twitch, and YouTube, and these platforms are used to spread some popular programmes. In particular, the live-soccer event applications [7] account for the large proportion of improving the people's enjoyment of life to a great extent, and the classical and well-known representatives are the Europe five major league matches [8] and FIFA world cup [9]. For the network in the smart cities, it is very necessary to recognize the live-soccer event application, which can enhance the network performance and improve the watching quality. However, the traditional Internet only provides the best-effort services and cannot do the differentiated scheduling for the special live-soccer event applications on the condition where the network resources in smart cities are limited [10]. On the other hand, the live-soccer events belong to the real-time applications, which have high requirements on delay, delay jitter, and packet loss rate. Since different applications have different requirements on Quality of Services (QoS) [11, 12] (for example, email applications have high requirements on packet loss rate and low requirements on delay and delay jitter), doing the application awareness and further making the differentiated scheduling are very significant. In other words, for the live-soccer event applications, they should be recognized in advance.

Furthermore, from the perspective of data transmission, the QoS-aware routing mechanism can be employed to provide differentiated services rather than the best-effort services [13]. In order to guarantee the QoS requirements of applications (e.g., the live-soccer events) during the process of data transmission, the Internet Engineering Task Force (IETF) gives three well-known models, i.e., Integrated Services (IntServ) [14], Differentiated Services (DiffServ) [15], and Multi-Protocol Label Switching (MPLS) [16]. Among them, the IntServ model uses resource reservation protocol to ensure that each node has the reserved resources, that is to say, all nodes have to store the network status related to the services. However, the functions of different devices are different (difficult to unify these configurations of devices), which causes bad scalability and high complexity. Different from IntServ, DiffServ divides network applications into several service ranks, and the different service rank has different handling method. Especially when the phenomena of network congestion happens, the related data traffic will be scheduled according to the service rank. However, DiffServ only guarantees the QoS requirements of applications for each service rank but cannot for the end-to-end QoS guarantee. In the MPLS model, the node forwards data according to the label. Although this model is simple, it is very hard to configure, manage, and debug for the involved devices.

With the above statements, the Software-Defined Networking (SDN) [17–19] has been accepted as a novel network paradigm and it is a feasible and efficient solution to address QoS awareness for the special live-soccer event applications. Different from the traditional network architecture, SDN separates the data plane from the control plane, i.e., managing the network system via the concentrated way. In addition, SDN can easily obtain the global network view and status information, such as flow statistics, the availability of network resources, the network topology information, and even the more detailed forwarding/routing information. On this basis, the control plane of SDN gives real-time control policies, and then, they are devolved to the data plane, which can guarantee the unified management and configuration for the network resources. In particular, the control plane of SDN can customize different control policies for different data flow according to the QoS requirements, in order to complete the fine-grained management. In other words, SDN can provide the differentiated end-to-end services for the special live-soccer event applications, greatly enhancing the network performance and improving the watching quality.

Therefore, we in this paper plan to employ SDN to make QoS awareness for the special live-soccer event applications in smart cities, in which the QoS-aware routing mechanism is proposed, called LSEA. The main contributions of this paper are summarized as follows. (1) SDN-based QoS-aware routing model is introduced, where delay, delay jitter, and packet loss rate are considered as three objects for optimization. (2) The improved Dijkstra routing algorithm and SDN-based disjoint routing algorithm are devised for the differentiated end-to-end services. (3) The proposed LSEA is implemented over Mininet to prove its feasibility and efficiency based on four metrics, i.e., delay, delay jitter, packet loss rate, and system recovery time.

The rest of the paper is organized as follows. Section 2 reviews the related work. In Section 3, the system model and three objective functions are quantified. Section 4 introduces the method of network status information collection. Section 5 devises the improved Dijkstra routing algorithm and SDN-based disjoint routing algorithm. Section 6 reports the experimental results, and finally, Section 7 concludes this paper.

2. Related Work

As we know, SDN-based QoS-aware routing mechanisms mainly include a single-path-based routing mechanism and a multiple-path-based routing mechanism.

There have been some single-path-based routing mechanisms guaranteeing the QoS requirements of applications. In [20], the OpenFlow switches were enabled to cooperate with the legacy switches by using the learning bridge protocol without requiring any modification on legacy switches. By utilizing the characteristics of SDN, SDN applications could dynamically find routing paths according to predefined QoS requirements and current network status. In [21], the use of QoS-based routing scheme over SDN was investigated, in which a real SDN test-bed is constructed by using Raspberry Pi computers as virtual SDN switches managed by a centralized controller. In [22], the model of adaptive routing of heterogeneous traffic with respect to the current QoS provisioning requirements was proposed, of which the main idea was to develop the model for effective routing with a high degree of flexibility achieved by using a set of weighting coefficients which all together constituted the general routing metrics. In [23], a server-driven bit rate estimation approach to compute the video bit rate and inform the application QoS requirement to the control layer was proposed. In addition, a QoS routing design for adaptive stream was devised, which allowed the SDN controller to evaluate all passable paths based on whole network topology by taking the bit rate of the segments into account.

Furthermore, some multiple-path-based routing mechanisms to guarantee the QoS requirements of applications have also been proposed. In [24], a new routing algorithm to calculate the bandwidth-delay constrained routes in a fast and efficient manner was presented. The algorithm was designed for the software-defined backbone networks, where the control plane was separated from the data plane and logically centralized. Besides providing the required QoS, the algorithm is aimed at maximizing the utilization of network resources. In [25], a source routing scheme to conduct the top-K QoS-aware paths discovery in SDN was introduced. First, the novel noninvasive QoS scheme was designed to collect QoS information based on LLDP in a piggyback fashion. Then, the variations of the K-shortest path algorithm were derived to find the unconstrained/constrained top-K ranked paths with regard to individual/overall path costs, reflecting QoS. In [26], the utilization of segment routing in SDN-based networks was explored to improve the network resource utilization and end users' QoS for delivering multimedia services over 5G networks. In [27], a mechanism to support on-demand QoS routing without any help from traffic engineering in SDN was proposed. By exploiting the fact that a controller could provide multiple routing paths based on global topology in SDN, the authors further proposed to monitor QoS over multiple routing paths and select the appropriate path satisfying users' QoS requirement. In [28], the end-to-end QoS based on the queue support in

OpenFlow was proposed, which allowed an operator with an SDN-enabled network to efficiently allocate the network resources according to the users' demands. Especially for each flow, the proposed solution guaranteed the required end-to-end QoS, while efficiently managing the utilization of open virtual switches.

As a matter of fact, the single-path routing can satisfy the QoS requirements due to the regulation of multiple weight factors. However, when the given path happens network congestion, it cannot provide the subsequent QoS provisioning services. Thus, in order to guarantee the reliability of routing, the multiple-path routing has been investigated. However, when the congestion phenomena of the common link(s) happen, a similar problem with respect to the single-path routing will emerge. Given such consideration, this paper plans to improve the Dijkstra routing algorithm and devise the SDN-based disjoint routing algorithm, guaranteeing the differentiated end-to-end services for the special live-soccer event applications.

3. Frame Structure

3.1. System Model. As depicted in Figure 1, the proposed LSEA is consisted of two major modules, i.e., network status information collection and path computing for the live-soccer events. Therein, the network status information collection module has two functions, i.e., topology management and network measurement, which is completed by the OpenFlow switch. The path computing module includes the improved Dijkstra routing and the SDN-based disjoint routing, which is used for the differentiated scheduling with respect to the special live-soccer event applications. In particular, the improved Dijkstra routing is used to obtain the multiple equivalent shortest paths in order to improve the response speed. The SDN-based disjoint routing is aimed at determining some disjoint paths from those equivalent shortest paths in order to improve the fault tolerance and the reliability of QoS routing. Meanwhile, the optimal path from these disjoint paths is selected to transmit the application data related to live-soccer events. To sum up, the improved Dijkstra routing is the foundation of SDN-based disjoint routing.

3.2. Optimization Objective. In order to learn the network status, more network information metrics should be collected. Based on the adequate information collection, the proper routing decision can be made well. Furthermore, in order to satisfy the various service requirements, this paper computes QoS routing with the multiple factors considered, balancing different factors. In particular, the multiple factors are originated from QoS parameters, including delay, delay jitter, and packet loss rate. However, the multifactor constraint QoS routing belongs to the NP-hard problem. Given this, we adopt the weighted evaluation function to transform the multiobjective model into a single-objective model, structuring a new routing measurement. In addition, different applications have different QoS requirements; thus, this paper adjusts the delay, delay jitter, and packet loss rate to change the objective function value and further satisfy the different QoS requirements. For example, email applications are impacted by the packet loss rate. We can adjust the weight value with respect to the packet loss rate to be large, so that these links with a high packet loss rate can be bypassed effectively.

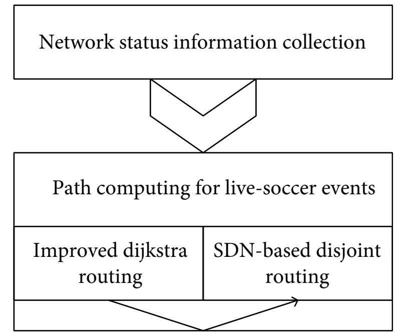

Figure 1: The frame structure of LSEA.

Given two arbitrary nodes R_i and R_j, let de_{ij}, jt_{ij}, and ls_{ij} denote the delay, the delay jitter, and the packet loss rate between R_i and R_j, respectively, and the single-objective function is defined as follows.

$$\text{minimize } f_{ij} = \alpha * de_{ij} + \beta * jt_{ij} + \gamma * ls_{ij}, \tag{1}$$

where α, β, and γ are the weights with respect to delay, delay jitter, and packet loss rate, respectively, and $\alpha + \beta + \gamma = 1$. However, the three metrics are not in the same order of magnitude; thus, the standardization operation should be performed.

Let de'_{ij}, jt'_{ij}, and ls'_{ij} denote the standardized results on de_{ij}, jt_{ij}, and ls_{ij}, respectively, and we have

$$de'_{ij} = \frac{de_{ij} - de_{\min}}{de_{\max} - de_{\min}}, \tag{2}$$

where $de_{\max}$ and $de_{\min}$ are the maximal and minimal link delay between R_i and R_j.

$$jt'_{ij} = \frac{jt_{ij} - jt_{\min}}{jt_{\max} - jt_{\min}}, \tag{3}$$

where $jt_{\max}$ and $jt_{\min}$ are the maximal and minimal link delay jitter between R_i and R_j.

$$ls'_{ij} = \frac{ls_{ij} - ls_{\min}}{ls_{\max} - ls_{\min}}, \tag{4}$$

where $ls_{\max}$ and $ls_{\min}$ are the maximal and minimal link packet loss rate between R_i and R_j.

4. Network Status Collection

The traditional network status information collection usually adopts the offline method since the network cannot automatically obtain the traffic information. As a result, it is very necessary to find an online method to complete the efficient QoS-aware routing for the special live-soccer event applications while there is no need to spend much time for collecting the statistical information. With such consideration, we use SDN to collect the network status information for all applications, because SDN has a global awareness function on the network status information without the additional overhead.

In terms of the network status information collection in SDN, some flow tables are requisite. However, these flow tables are only installed by the OpenFlow protocol [29, 30]; therefore, SDN needs the help of the OpenFlow switch, that is to say, this paper deploys OpenFlow switch to help complete the network status information collection under SDN environment.

Based on the above discussion, the structure of network status information collection is shown in Figure 2, including the SDN network environment and the traditional Internet environment. Meanwhile, the communication between the SDN network and the traditional Internet depends on two switches: ordinary switch and OpenFlow switch. Regarding the network status information collection based on two switches, the related process is described as follows. At first, the ordinary switch collects the network status information of all users. Then, the ordinary switch submits the collected network status information to the OpenFlow switch by the techniques of port mirroring and redirection, just like the data replication. Finally, the OpenFlow switch sends the related network status information to its corresponding SDN controller.

5. QoS-Aware Path Computing

This section will make the path computing, i.e., realizing the QoS-aware routing mechanism for the special live-soccer event applications in smart cities, including the improved Dijkstra routing and the SDN-based disjoint routing. Meanwhile, the former is used to obtain the multiple equivalent shortest paths in order to improve the response speed, while the latter is to select two or three disjoint paths from those equivalent shortest paths in order to improve the fault tolerance and the reliability of QoS routing.

5.1. Improved Dijkstra Routing. The original Dijkstra algorithm computes all shortest paths with respect to any two nodes, which cannot satisfy the requirements of allocating different QoS ranks for different applications because of two major limitations, as follows.

(i) It computes all shortest paths regarding any two nodes, which increases the time computation overhead

(ii) It only computes the shortest path between any two nodes, irrespective of the computing of multiple shortest paths between any two nodes, because the data structure to store the front node is only an array (only record a front node in terms of the current node) and cannot store the multifront nodes.

In order to satisfy the requirements of different QoS rank allocation, the multiple equivalent shortest paths should be selected in advance. Given this, we improve the Dijkstra routing algorithm based on the following two aspects.

(i) When the system finds the destination node, the routing algorithm is finished, which can decrease the time computation overhead.

(ii) The data structure of storing the front node is modified from the array to the linked list.

Based on the above statements, the improved Dijkstra routing algorithm is described in Algorithm 1, in which the new measurement value on the link is updated according to equation (1).

Among them, T_nodes is the total number of nodes, pre_nodes is the set of front nodes regarding the current node, dist is the distance between source node to the destination node, detec is the set of nodes which have been detected, and undetec is the set of nodes which have not been detected.

5.2. SDN-Based Disjoint Routing. For one path, the smaller remaining bandwidth means the larger probability to happen network congestion. Here, the remaining bandwidth of the path is defined as the minimal bandwidth value among all involved links with respect to the current path. As the above mentioned, SDN can have a good command of knowledge on the global network view. Therefore, the SDN controller can measure the remaining bandwidth information periodically. Furthermore, in order to avoid network congestion, this paper considers the path that has the maximal remaining bandwidth as the optimal path to transmit the live-soccer event applications, which can guarantee the QoS requirements, improve the throughput and response time, and complete the load balance.

Moreover, some disjoint paths from those equivalent shortest paths obtained by the improved Dijkstra routing algorithm are selected to guarantee the fault tolerance and the reliability of QoS routing, where the remaining bandwidth of the path is regarded as the evaluation metric. In particular, the path with the maximal remaining bandwidth is used for the main routing, while the rest of the paths are used for the alternative routing.

For example, in Figure 3, the cost of the link refers to the bandwidth. Given two paths from node A to D, i.e., A-B-D and A-B–C-D. The first path's remaining bandwidth is 1 ($\min\{3, 1\}$), and the second path's remaining bandwidth is 2 ($\min\{3, 4, 2\}$). As a conclusion, the second path is regarded the main routing.

The SDN-based disjoint routing is composed of the following four operations. At first, the path is converted into some links. Then, according to the link's remaining bandwidth information, the path's remaining bandwidth is updated through the form of iteration. Thirdly, these paths are arranged in the descending order according to the path's remaining bandwidth. Finally, the path with the maximal remaining bandwidth is determined in advance, and the correspondingly involved links constitute some disjoint paths. Based on the above statements, the detailed SDN-based disjoint routing algorithm is described in Algorithm 2.

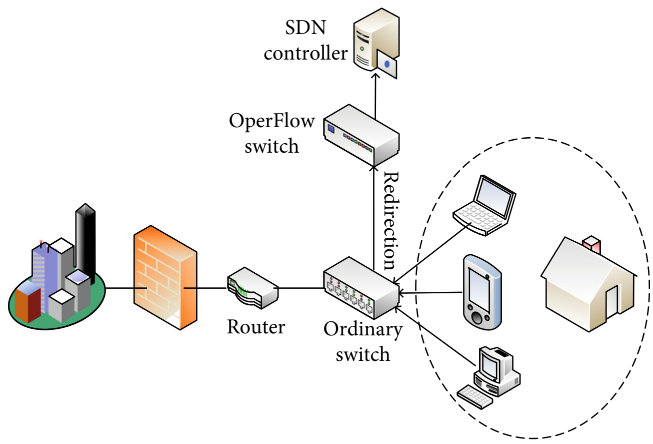

FIGURE 2: The structure of network status information collection.

Initialize T_nodes, pre_nodes, dist, detec, undetec, and Path;
Update the new measurement value according to equation (1);
Store the sour to detec;
while undetec ≠ Null and node ≠ dest, **do**
Select node according to the shortest dist;
Update dist;
Update pre_nodes;
endwhile
Compute the multiple equivalent shortest paths according to pre_nodes;
Update Path;
return Path;

ALGORITHM 1: The improved Dijkstra routing.

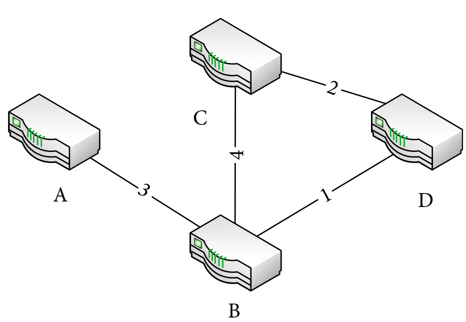

FIGURE 3: An example to illustrate the determination of the main routing.

Among them, minbw is the set used to store the minimal remaining bandwidth for each path, P_to_L is the set used to store all involved links in terms of the current path; disjP is the set used to store the obtained disjoint paths, and pri_ min _bw is the minimal remaining bandwidth to which disjP corresponds.

Particularly, from the perspective of time complexity, Algorithm 1 runs in $O(n^2 \log n)$ because it only computes the shortest path between the source node and destination node; Algorithm 2 runs in $O(n \log n)$ because SDN has the global network view and the main computation overhead comes from the arrangement operation.

6. LSEA Evaluation

6.1. Environment Setup. The proposed LSEA in smart cities is implemented based on the Intel(R) Core(TM) i5-8500 CPU @3.00GHz, RAM 8.00GB, running on the Ubuntul6.02 64bits operation systems. The programming language is Python, running on the Pycharm. The SDN network environment is implemented based on Mininet [31], where the Ryu controller is employed and the Iperf instrument is used to send packets. The simple process is described as follows. At first, the SDN network environment is created over Mininet; secondly, the network environment is connected with the remote Ryu controller; thirdly, two hosts are opened: one is regarded as the source and the other one is regarded as the destination, in which the Iperf instrument is started to send packets; finally, the path is computed to forward the live-soccer event applications.

6.2. Data Collection. At present, there are no open datasets on delay, delay jitter, loss packet rate, and bandwidth information; thus, this paper uses some programs to simulate the similar dataset information, including QoS parameters. At first, we use files to store the network topology: if two nodes are adjacent, the corresponding status is expressed by 1; otherwise, the corresponding status is expressed by 0. Then, the file of network topology is read, generating the related QoS information. Finally, three metrics, i.e., delay, delay

```
Initialize minbw, P_to_L, disjP and pri_ min _bw;
Put Path into P_to_L;
For each path in Path, do
Update minbw;
minbw = newminbw;
endfor
Arrange these paths in Path according to minbw;
Compute disjP according to minbw;
Obtain pri_ min _bw according to disjP;
return disjP;
```

Algorithm 2: The SDN-based disjoint routing.

jitter, and loss packet rate, are standardized to guarantee that they are at the same order of magnitude.

6.3. Simulation Setting. The Deltacom network topology with 97 nodes and 124 links is used for simulation, as shown in Figure 4, where there are three requesters and four providers. The delay, delay jitter, loss packet rate, and system recovery time are considered as four performance evaluation metrics. In addition, two baselines, i.e., single-path-based QoS routing mechanism [23] and multiple-path-based QoS routing mechanism [27], shorten for ISPCS and NSW, respectively, are selected as the comparison benchmarks. Three parameters, α, β, and γ, are randomly set as 0.3, 0.4, and 0.3, respectively. For each number of packets, we make 100 times simulations, that is to say, the average experiment results are based on 100 times simulations.

6.4. Results

6.4.1. Delay. The experimental results on delay are shown in Figure 5. We can find that the average delays of LSEA, ISPCS, and NSW are 80.26 ms, 89.38 ms, and 84.38 ms, respectively; that is to say, the proposed LSEA has the obvious advantage on the average delay. In fact, the weight of delay in LSEA is larger than those in both ISPCS and NSW, which can filter out some links with the relatively large delay. Thus, the end-to-end delay of LSEA is smaller than those of ISPCS and NSW.

6.4.2. Delay Jitter. The experimental results on delay jitter are shown in Figure 6. We can find that the average delay jitters of LSEA, ISPCS, and NSW are 10.63 ms, 13.17 ms, 15.12 ms, respectively. In other words, LSEA has smaller delay jitter than ISPCS and NSW; this is because LSEA filters out some links with the relatively large delay jitter via the setting of β. In addition, the SDN-based disjoint routing mechanism can guarantee the stable transmission of live-soccer event applications. However, ISPCS and NSW do not consider disjoint routing.

6.4.3. Loss Packet Rate. The experimental results on the loss packet rate are shown in Figure 7. We can find the average loss packet rates of LSEA, ISPCS, and NSW are 0.042, 0.038, and 0.041, respectively. Although the proposed LSEA has a relatively larger loss packet rate than ISPCS and NSW, the corresponding results are very close. Based on such consideration, we think that the proposed LSEA is acceptable in terms of the loss packet rate.

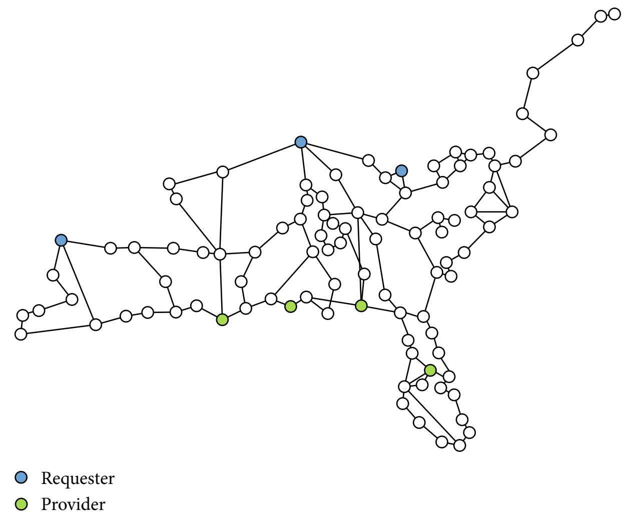

Figure 4: Deltacom network topology used for simulation.

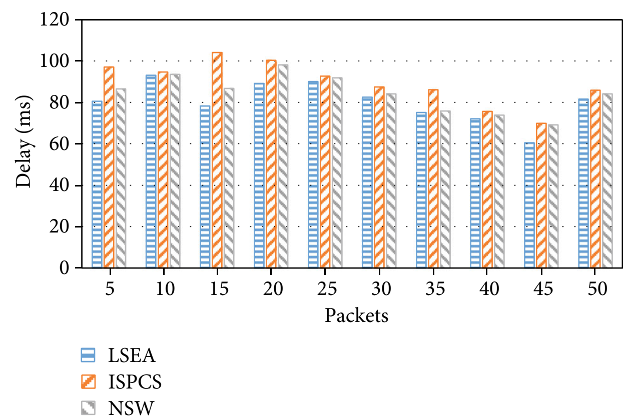

Figure 5: The experimental results on the average delay.

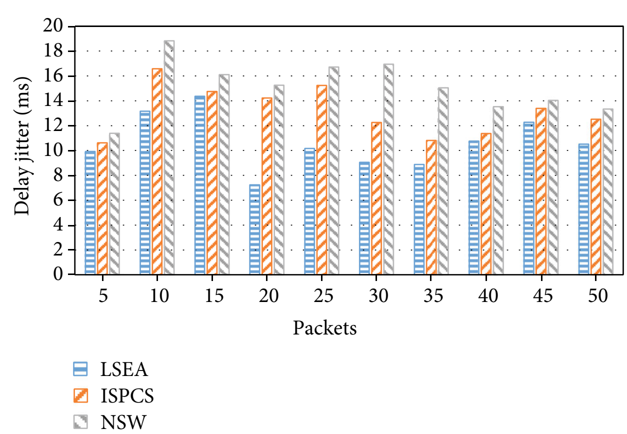

Figure 6: The experimental results on the average delay jitter.

6.4.4. System Recovery Time. In this section, we disconnect two or three links periodically, and the time period is set as 5 s. The experimental results on system recovery time are shown in Figure 8. We can find that the proposed LSEA

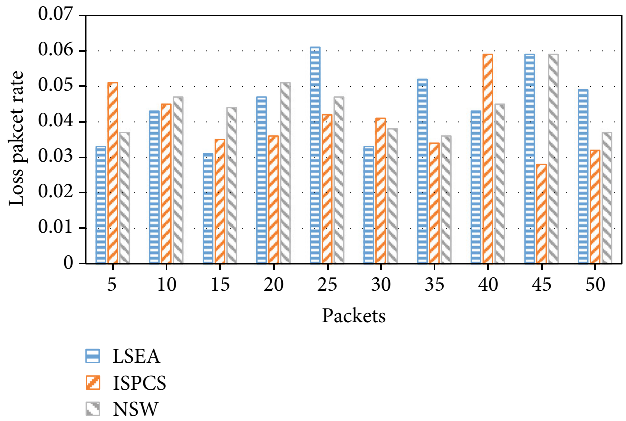

FIGURE 7: The experimental results on the average loss packet rate.

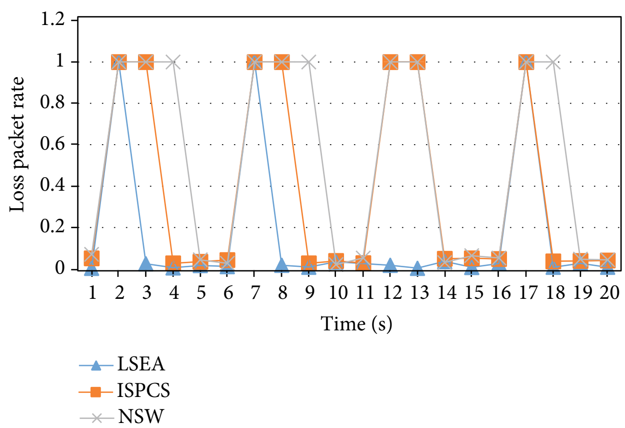

FIGURE 8: The experimental results on the average system recovery time.

always can recover the system stability with the fastest speed. Two reasons are listed as follows. On the one hand, LSEA has some disjoint paths, which can provide the transmission guaranteeing for the special live-soccer event applications. On the other hand, LSEA uses SDN to control the network in a centralized manner, which can schedule the feasible path for the live-soccer event applications in case of link failure.

7. Conclusions

With the rapid development of new Internet technologies such as mobile edge computing, 5G, and IoT, the building of smart cities has attracted much attention from many countries and regions, which can provide a better connection between services and citizens based on the combination of these new Internet technologies. For the special live-soccer event applications in the smart cities, this paper employs SDN to make QoS-awareness routing. At first, the SDN-based QoS-aware routing model is introduced, where delay, delay jitter, and packet loss rate are considered as three objects for optimization. Then, the improved Dijkstra routing algorithm and SDN-based disjoint routing algorithm are devised for the differentiated end-to-end services. In particular, the improved Dijkstra routing is the foundation of SDN-based disjoint routing. Finally, the proposed LSEA is implemented over Mininet to prove its feasibility and efficiency based on delay, delay jitter, packet loss rate, and system recovery time.

However, this paper also has some limitations. On the one hand, the analysis of time complexity and space complexity is not included. On the other hand, more applications should be employed to make the simulation experiments. In the future, we will do a further study on LSEA around the abovementioned two limitations. Furthermore, we also plan to improve LSEA based on the method of machine learning.

Abbreviations

DiffServ: Differentiated services
IETF: Internet Engineering Task Force
IntServ: Integrated Services
IoT: Internet of Things
ISPCS: Intelligent signal processing and communication systems [23]
LSEA: The proposed routing mechanism in this paper
MPLS: Multi-Protocol Label Switching
NSW: Network softwarization and workshops [27]
QoS: Quality of Service
SDN: Software-defined networking.

Data Availability

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

Conflicts of Interest

None of the authors have any conflicts of interest.

Acknowledgments

We thank Ping Li, who is a senior experimenter to provide the SDN network environment.

References

[1] H. Li, G. Shou, Y. Hu, and Z. Guo, "Mobile edge computing: progress and challenges," in *2016 4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, pp. 83-84, Oxford, UK, March 2016.

[2] P. T. Dat, A. Kanno, N. Yamamoto, and T. Kawanishi, "5G transport networks: the need for new technologies and standards," *IEEE Communications Magazine*, vol. 54, no. 9, pp. 18–26, 2016.

[3] Y. Liu, K. Wang, K. Qian, M. Du, and S. Guo, "Tornado: enabling blockchain in heterogeneous Internet of Things through a space-structured approach," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1273–1286, 2020.

[4] S. Sengupta and S. S. Bhunia, "Secure data management in cloudlet assisted IoT enabled e-Health framework in smart city," *IEEE Sensors Journal*, vol. 20, no. 16, pp. 9581–9588, 2020.

[5] E. J. Cedillo-Elias, J. A. Orizaga-Trejo, V. M. Larios, and L. A. M. Arellano, "Smart government infrastructure based in SDN networks: the case of guadalajara metropolitan area," in *2018*

IEEE International Smart Cities Conference (ISC2), pp. 1–4, Kansas City, MO, USA, September 2018.

[6] Y. Li, W. Ren, T. Zhu, Y. Ren, Y. Qin, and W. Jie, "RIMS: a real-time and intelligent monitoring system for live-broadcasting platforms," *Future Generation Computer Systems*, vol. 87, pp. 259–266, 2018.

[7] X. Yu, L. Li, and H. W. Leong, "Interactive broadcast services for live soccer video based on instant semantics acquisition," *Journal of Visual Communication and Image Representation*, vol. 20, no. 2, pp. 117–130, 2009.

[8] V. Hofer and J. Leitner, "Relative pricing of binary options in live soccer betting markets," *Journal of Economic Dynamics and Control*, vol. 76, pp. 66–85, 2017.

[9] J. L. Nicolau and A. Sharma, "A generalization of the FIFA World Cup effect," *Tourism Management*, vol. 66, pp. 315–317, 2018.

[10] U. Gulec, M. Yilmaz, V. Isler, R. V. O'Connor, and P. M. Clarke, "A 3D virtual environment for training soccer referees," *Computer Standards & Interfaces*, vol. 64, pp. 1–10, 2019.

[11] C. Abid, M. Kessentini, and H. Wang, "Early prediction of quality of service using interface-level metrics, code-level metrics, and antipatterns," *Information and Software Technology*, vol. 126, article 106313, 2020.

[12] J. O. Mebawondu, F. M. Dahunsi, and O. S. Adewale, "Hybrid intelligent model for real time assessment of voice quality of service," *Scientific African*, vol. 9, article e00491, 2020.

[13] Q. He, J. Yan, H. Jin, and Y. Yang, "Quality-aware service selection for service-based systems based on iterative multi-attribute combinatorial auction," *IEEE Transactions on Software Engineering*, vol. 40, no. 2, pp. 192–215, 2014.

[14] Z. Shan, "Integrated service adaptation," in *2010 6th World Congress on Services*, pp. 140–143, Miami, FL, USA, July 2010.

[15] Y. Wang, X. Wang, H. Li, Y. Dong, Q. Liu, and X. Shi, "A multi-service differentiation traffic management strategy in SDN cloud data center," *Computer Networks*, vol. 171, article 107143, 2020.

[16] Z. Al-Qudah, I. Jomhawy, M. Alsarayreh, and M. Rabinovich, "On the stability and diversity of Internet routes in the MPLS era," *Performance Evaluation*, vol. 138, article 102084, 2020.

[17] M. Karakus and A. Durresi, "Quality of service (QoS) in software defined networking (SDN): a survey," *Journal of Network and Computer Applications*, vol. 80, pp. 200–218, 2017.

[18] A. A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, "5G network slicing using SDN and NFV: a survey of taxonomy, architectures and future challenges," *Computer Networks*, vol. 167, article 106984, 2020.

[19] R. Amin, M. Reisslein, and N. Shah, "Hybrid SDN networks: a survey of existing approaches," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3259–3306, 2018.

[20] C. Lin, K. Wang, and G. Deng, "A QoS-aware routing in SDN hybrid networks," *Procedia Computer Science*, vol. 110, pp. 242–249, 2017.

[21] A. Kucminski, A. Al-Jawad, P. Shah, and R. Trestian, "QoS-based routing over software defined networks," in *2017 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pp. 1–6, Cagliari, Italy, June 2017.

[22] M. Beshley, M. Seliuchenko, O. Panchenko, and A. Polishuk, "Adaptive flow routing model in SDN," in *2017 14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)*, pp. 298–302, Lviv, Ukraine, 2017.

[23] X. Jin, H. Ju, S. Cho, B. Mun, C. Kim, and S. Han, "QoS routing design for adaptive streaming in software defined network," in *2016 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, pp. 1–6, Phuket, Thailand, October 2017.

[24] S. Tomovic and I. Radusinovic, "Fast and efficient bandwidth-delay constrained routing algorithm for SDN networks," in *2016 IEEE NetSoft Conference and Workshops (NetSoft)*, pp. 303–311, Seoul, South Korea, June 2016.

[25] X. Chen, J. Wu, and T. Wu, "The top-K QoS-aware paths discovery for source routing in SDN," *KSII Transactions on Internet & Information Systems*, vol. 12, no. 6, pp. 2534–2553, 2018.

[26] A. Barakabitze, L. Sun, I. Mkwawa, and E. Ifeachor, "A novel QoE-centric SDN-based multipath routing approach for multimedia services over 5G networks," in *2018 IEEE International Conference on Communications (ICC)*, pp. 1–7, Kansas City, MO, USA, May 2018.

[27] T. Kim and T. Nguyen-Duc, "OQR: on-demand QoS routing without traffic engineering in software defined networks," in *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*, pp. 362–365, Montreal, QC, Canada, June 2018.

[28] D. L. C. Dutra, M. Bagaa, T. Taleb, and K. Samdanis, "Ensuring end-to-end QoS based on multi-paths routing using SDN technology," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pp. 1–6, Singapore, Singapore, December 2017.

[29] E. L. Fernandes, E. Rojas, J. Alvarez-Horcajo et al., "The road to BOFUSS: the basic OpenFlow userspace software switch," *Journal of Network and Computer Applications*, vol. 165, article 102685, 2020.

[30] R. Jmal and L. C. Fourati, "An OpenFlow architecture for managing content-centric-network (OFAM-CCN) based on popularity caching strategy," *Computer Standards & Interfaces*, vol. 51, pp. 22–29, 2017.

[31] Mininethttp://mininet.org.