Hindawi
Mobile Information Systems
Volume 2021, Article ID 5595260, 10 pages
https://doi.org/10.1155/2021/5595260

Research Article

Intelligent Forwarding Strategy for Congestion Control Using Q-Learning and LSTM in Named Data Networking

Sanguk Ryu,[1] **Inwhee Joe**,[1] **and WonTae Kim**[2]

[1]*Department of Computer and Software, Hanyang University, 222 Wangsimni-ro, Seoul 04763, Republic of Korea*
[2]*Department of Computer Science and Engineering, Korea University of Technology and Education, Cheonan 31253, Republic of Korea*

Correspondence should be addressed to WonTae Kim; wtkim@koreatech.ac.kr

Received 7 February 2021; Revised 9 March 2021; Accepted 23 March 2021; Published 31 March 2021

Academic Editor: Jianhui Lv

Named data networking (NDN) is a future network architecture that replaces IP-oriented communication with content-oriented communication and has new features such as cache, multiple paths, and multiple sources. Services such as video streaming, to which NDN can be applied in the future, can cause congestion if data is concentrated on one of the nodes during high demand. To solve this problem, sending rate control methods such as TCP congestion control have been proposed, but they do not adequately reflect the characteristics of NDN. Therefore, we use reinforcement learning and deep learning to propose a congestion control method that takes advantage of multipath features. The intelligent forwarding strategy for congestion control using Q-learning and long short-term memory in NDN proposed in this paper is divided into two phases. The first phase uses an LSTM model to train a pending interest table (PIT) entry rate that can be used as an indicator to detect congestion by knowing the amount of data returned. In the second phase, it is forwarded to an alternative path that is not congestive via Q-learning based on the PIT entry rate predicted by the trained LSTM model. The simulation results show that the proposed method increases the data reception rate by 6.5% and 19.5% and decreases the packet drop rate by 7.3% and 17.2% compared to an adaptive SRTT-based forwarding strategy (ASF) and BestRoute.

1. Introduction

The rapid development of the Internet has led to a significant increase in the amount of content transmitted every year. However, these changes have been difficult to adapt to because the current Internet architecture depends on IP addresses and is designed for end-to-end communication. This drawback causes problems such as transport efficiency and security.

Information-centric networking (ICN) was proposed as a solution to the problem caused by the rapid increase of content. The goal was to change the communication paradigm from a content-oriented model to the IP-based model [1]. Named data networking (NDN), one of the most well-known ICN architectures, is attracting attention as a hotspot for research [2–4].

NDN is a future network architecture that alters the current IP-based Internet as the Internet environment changes, replacing IP addresses with named content for communication. Compared with traditional TCP/IP, it has the following new features in the transmission method. First, NDN communication is a consumer-driven pull mode and is connectionless. The consumer sends an interest packet to request the content, and the producer with the requested content returns the matched data. The second is a multi-source feature. NDN has a content store (CS), where the returned content can be temporarily stored in the intermediate nodes in a network. Therefore, the consumer can receive the requested data from multiple sources, including the CS of the intermediate node and the producer with the original data. Third, NDN has multipath features, so it supports dynamic multipath forwarding. The NDN node provides multiple paths from the consumer to sources via a forwarding information base (FIB) that stores the interface information where the packet can go to next. It then decides

how to use the path provided through a forwarding strategy. Although this change solves the limitations of the current Internet to some extent, if NDN is applied to a service such as video streaming, congestion may occur at a node, where data is concentrated when people are crowded during a certain period. Therefore, congestion control is one of the major research tasks of NDN.

Congestion control of NDN has been proposed by applying the TCP/IP method. TCP/IP congestion control detects congestion via the retransmission timeout (RTO) and adjusts the sending rate via an additive increase/multiplicative decrease (AIMD) window-based mechanism. However, congestion detection through RTO is not a reliable indicator in NDN, where different round-trip times (RTTs) are measured for each source as it has a multisource feature. Furthermore, the window control method targeting a single path of TCP/IP is not suitable for NDN due to its characteristics of multiple sources and multiple paths. The reason is that when a consumer receives data from two sources through different paths, if one path is congested and the consumer reduces the window size, the throughput of the other path that is not congested also decreases. As such, the direct application of existing solutions does not adequately consider the characteristics of NDN, so network congestion control methods must also change. Therefore, it is necessary to propose a new congestion control method for NDN.

In this paper, we propose an intelligent forwarding strategy for congestion control using Q-learning and LSTM in named data networking (IFS-QLSTM) using a dynamic forwarding method to utilize multiple paths. First, the IFS-QLSTM uses the LSTM model to train the number of entries that change due to packets added to the pending interest table (PIT) in the NDN node (we use the term PIT entry rate interchangeably in the rest of the paper). Second, the PIT entry rate predicted by the trained LSTM model is used for the reinforcement learning to judge the congested node. The node is then bypassed and the packet is forwarded.

The rest of this paper is organized as follows. Section 2 explains the background of NDN and related research. Section 3 describes an intelligent forwarding strategy for congestion control using Q-learning and LSTM in NDN. Section 4 presents the performance evaluation and analysis of the results through simulation. Finally, Section 5 concludes the paper.

2. Related Works

In recent years, NDN has been studied as a future network architecture that will replace the current Internet. One of the core technologies of NDN architecture is congestion control. We survey related studies in two aspects: (1) studies on control of the interest sending rate for congestion control and (2) studies on adaptive forwarding strategy [5–7].

Researches on the interest sending rate for congestion control include a receiver-based window control method and a hop-by-hop interest sending rate control method. In [8], the authors describe a receiver-based window control scheme that controls the interest sending rate by adjusting the congestion window using a TCP-like mechanism in the receiver of RTT. Similarly, both ICTP and CCTCP use a method of adjusting the congestion window based on RTT [9, 10]. However, the NDN caches data through the CS added to the router, which causes the RTT to change irregularly. In addition, when a consumer requests data from multiple sources and one source is congested, the consumer reduces the window size. This means that the amount of transmission to the source where congestion does not occur is also reduced. Therefore, the traditional receiver-based window control congestion control method is not suitable for NDN. In [11], the authors demonstrate a representative hop-by-hop method that detects congestion in intermediate nodes and adjusts the interest sending rate using interest shaping. Wang et al. [12] proposed a method that improves [11] by adding NACK feedback to inform the downstream nodes of congestion.

A forwarding strategy can dynamically select one or more interfaces in the FIB to forward the interest packet. The BestRoute strategy forwards the interest packet using the path available at the lowest routing cost [13]. In [14], the authors propose a forwarding strategy based on calculating the weight value of the number of pending interests corresponding to each output interface of FIB. In [15], the authors design adaptive forwarding to retrieve data through optimally performing paths, quickly detect, and recover from packet transmission problems. In [16], the authors propose an adaptive SRTT-based forwarding strategy (ASF). The ASF periodically measures the SRTT of an adjacent node at each node, arranges the transmittable nodes based on this, selects the node with the lowest SRTT, and transmits the interest packet. If a problem such as a timeout occurs, the node in which the problem occurs is penalized and sent to the end of the sequence.

In this paper, we design an intelligent forwarding strategy for congestion control using Q-learning and LSTM in NDN. In the first phase, we predict the change in the PIT entry rate in the next time step through time series prediction based on a pretrained LSTM model. In the second phase, based on the predicted PIT entry rate, an appropriate alternative route is selected through Q-learning in congestion situations.

3. Proposed Method: IFS-QLSTM

3.1. Basic NDN Forwarding Mechanism. The NDN node is composed of three elements: PIT, CS, and FIB. The PIT records where the interest packet originated from when it came into the node and tells where to return the data packet when it comes in. CS is a place to temporarily store data and is a feature of the NDN nodes. FIB is a place where nodes that can go to each prefix are recorded, and when an interest packet comes in, it searches for the prefix and informs the path to go next.

Figure 1 illustrates the forwarding process of NDN. When an interest packet arrives, the NDN node first searches for the CS and then returns it to the incoming interface if there is matching data. If not, it goes to PIT and lookup. If duplicated data is already requested in the PIT, the path on which the interest packet came in is added. However, if not,

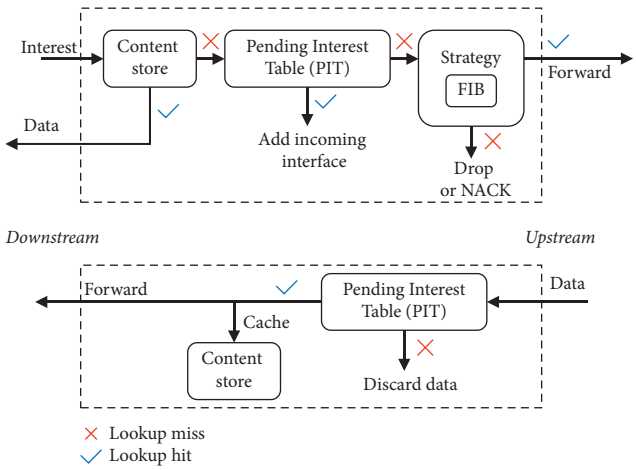


Figure 1: Interest and data processing in NDN node.

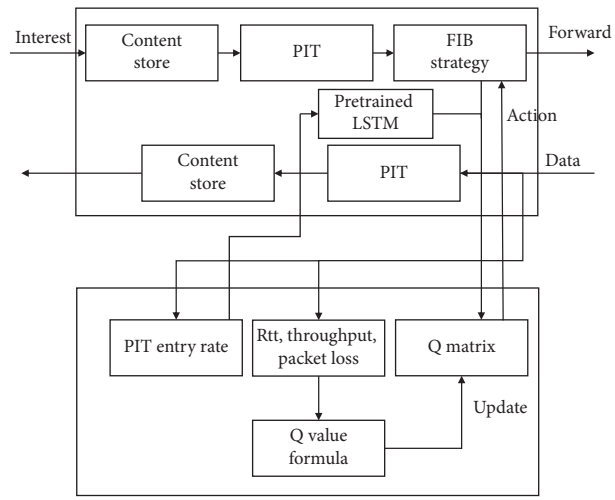


Figure 2: System model of the IFS-QLSTM.

it is recorded in the PIT and sent to the FIB. Finally, in FIB, if there is a node that can search for the name of a received packet and transmit it, it transmits it to the optimal path according to the forwarding strategy. However, if there is no transmittable node, the packet is discarded. Next, when a data packet arrives at the NDN node, it first searches the PIT and checks whether there is a request for the received data. If there is a request, it returns through the recorded reverse path, and if there is no request, the incoming data packet is discarded. Data that comes in before being transmitted over the reverse path is retrieved from the CS, and if there is no cached data, it is stored in the CS so that it can quickly respond to the next request.

3.2. Proposed System Model. The system model of an intelligent forwarding strategy for congestion control using Q-learning and LSTM in named data networking is shown in Figure 2. When the NDN node receives the interest packet, it checks whether there is a matching name in CS and PIT, and if not, the FIB searches the outgoing interface and forwards it to the interface chosen by the forwarding strategy. As shown in Figure 2, the IFS-QLSTM proceeds in the same way up to the PIT but shows the difference in the forwarding strategy to bypass the congested nodes. First, the PIT entry rate of neighboring nodes is predicted through pretrained LSTM using the PIT entry rate of the nodes obtained from the data packet. After that, congestion is detected using the predicted value as the state of Q-learning, and an appropriate alternative path is selected as the action and forwarded.

3.3. Pretrained LSTM. NDN's PIT is a place to record the incoming interface of the received interest packet, so it can predict the amount of returned data. Since it changes with time, it can also be viewed as time-series data. Thus, if we train using the LSTM model, a deep learning that is widely used for predicting time series data, we can predict the new PIT entry rate in the next time interval. Based on this data, it is possible to know the arrival rate of data packets, and the congestion can then be forecast in a timely manner.

In advance, the PIT entry rate for each node is measured and normalized to use as an input to the LSTM model. Then, as shown in Figure 3, we train the LSTM model to predict time $t+n+1$ by inputting time t through $t+n$. Finally, the trained model is saved and used to predict the next time step PIT entry rate of the neighboring nodes.

3.4. Q-Learning Structure

3.4.1. State. Reinforcement learning agents must be given enough information to accurately know their current state. However, in the case of the Q-learning used in this paper, if you use too much state to generate the q-table consisting of states and actions, it may cause problems with the q-table by becoming too complicated. Therefore, it has to choose an appropriate state variable that can represent the current state. In this paper, it shows the two following state variables: First, it is necessary to know where to make a decision, so the current node that has received the interest packet is set as the state. Second, to know the congestion condition of the nodes that can be transmitted by the current node, the predicted value of the PIT entry rate of the transmittable nodes using a pretrained LSTM model is set as the state. Based on these two states, it is possible to know where the agent is currently located and the congestion condition of the neighboring nodes.

3.4.2. Action. Since the IFS-QLSTM is a method of transmission by selecting an appropriate path for a congestion condition, when the NDN node receives an interest packet, one of the neighboring nodes that can be transmitted is selected as an action.

3.4.3. Reward. Since the reward is an indicator of the direction of training, the definition of reward is important in reinforcement learning. Therefore, to train in the desired direction, it is necessary to define a reward suitable for the training direction. Thus, the reward is defined as follows:

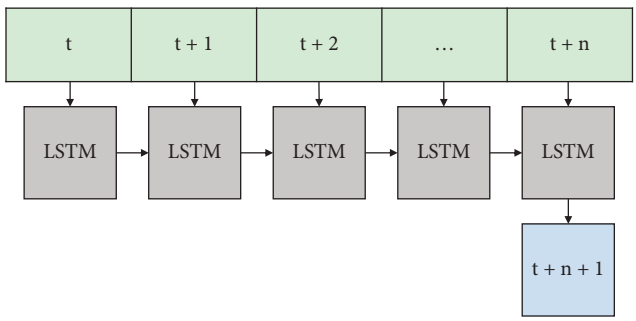


Figure 3: Pretrained LSTM

$$\text{reward} = \alpha \cdot \text{throughput}_n - \beta \cdot \text{packet loss}_n - \gamma \cdot \text{RTT}_n, \tag{1}$$

where N represents a node, and α, β, and γ are the weight values for controlling the throughput, packet loss, and RTT, respectively. Throughput represents the number of packets processed per second by node N, and packet loss represents the number of packets discarded per second by node N. RTT represents the time when a packet is transmitted and received by node N. We thought that if we set only the packet loss as a reward, it may be trained not to consider packet transmission time or throughput, although congestion paths were well avoided. Therefore, we designed the reward in the direction of increasing packet throughput and reducing packet loss and RTT while avoiding congested paths.

3.4.4. Q Value Update. In this paper, the Q value was updated every second. The update formula of the Q value is the general Q-learning update formula as shown in Equation (2). *Q(s, a)* represents the Q value when action A is performed in state S. The value of r is the reward when action A is taken in state S. The discount factor, γ, is a number between 0 and 1 which has the effect of valuing rewards received earlier as higher than those received later.

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left(r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right). \tag{2}$$

3.5. Q-Learning-Based Forwarding Strategy. Figure 4 shows the Q-learning packet transmission process when an interest packet is received by the NDN node. When the interest packet arrives, the NDN node first checks the CS and PIT for a matching name; if a matching name does not exist, it looks up the name in the FIB. If there is a matching name in the FIB, the PIT entry rate of the nodes corresponding to the matching name (transmittable nodes from the current node) is predicted using the pretrained LSTM. If not, the interest packet is discarded. After that, it is forwarded to the most optimal path through Q-learning. Specifically, the predicted PIT entry rate and the current node are used as the state of Q-learning to obtain the Q values of the transmittable nodes from the q table. Next, a random value between 0 and 1 is selected, and if it is less than the current epsilon value, the reinforcement learning agent selects the exploration method. The exploration method selects a random node among the remaining nodes except for the node with the highest Q value and forwards the interest packet. The reason for the exploration is that as the path that was not good in the past may improve, always making the optimal decision may not be good for reinforcement learning training, it is a method used to gain various experiences. Next, if the random value is greater than the epsilon value, the exploitation method is selected. This method selects the node with the largest Q value among the transmittable nodes in the q table and forwards the interest packet. In this way, exploration and exploitation are performed according to the epsilon value, but if the exploration is excessive, the performance is reduced, so the epsilon value is set to decrease over time.

4. Simulation and Analysis

4.1. Simulation Environment. In this section, we implemented by using the open-source ndnSIM [17, 18], an NS-3 based simulator that was developed for NDN. We then evaluated the performance of the IFS-QLSTM through simulation results. Two evaluation metric criteria were selected to quantitatively evaluate the effectiveness of our method. The first criterion was the rate of InData as an indicator for evaluating the utilization of the bottleneck links and alternate links. InData represents the amount of incoming data in the node and guarantees that this amount of data packets was actually transmitted during the congestion. The second criterion is the packet drop rate. If the packet drop rate of IFS-QLSTM is low, it can be seen that IFS-QLSTM effectively mitigates packet dropping.

The topology used in the experiment is shown in Figure 5. In the topology, the consumer (Node0) forwards an interest packet, and the producer (Node8) returns data matching the requested interest packet. The link bandwidth and delay in this topology are set to 10 Mbps and 10 ms, respectively. In our experiment, we cause congestion by setting a specific link bandwidth as low as 1 Mbps according to the requirements of various congestion scenarios.

Next, the Q-learning parameters of the IFS-QLSTM are as follows. First, a random variable (between 0.0 and 1.0) was assigned for comparison with epsilon. The epsilon value, which determines exploration and exploitation, decreased with time until it reached 0.01. The discount factor, which is the weight to control the future compensation compared to the current compensation, was set to 0.9. In the case of LSTM, Adam was used as the optimizer, and the learning rate was set to 0.001. We chose BestRoute and ASF because BestRoute is a basic NDN forwarding method used as a comparison algorithm in many papers, ASF is a more advanced forwarding algorithm, and the main reason is that both methods are verified algorithms. Therefore, we simulated them and compared them with the IFS-QLSTM.

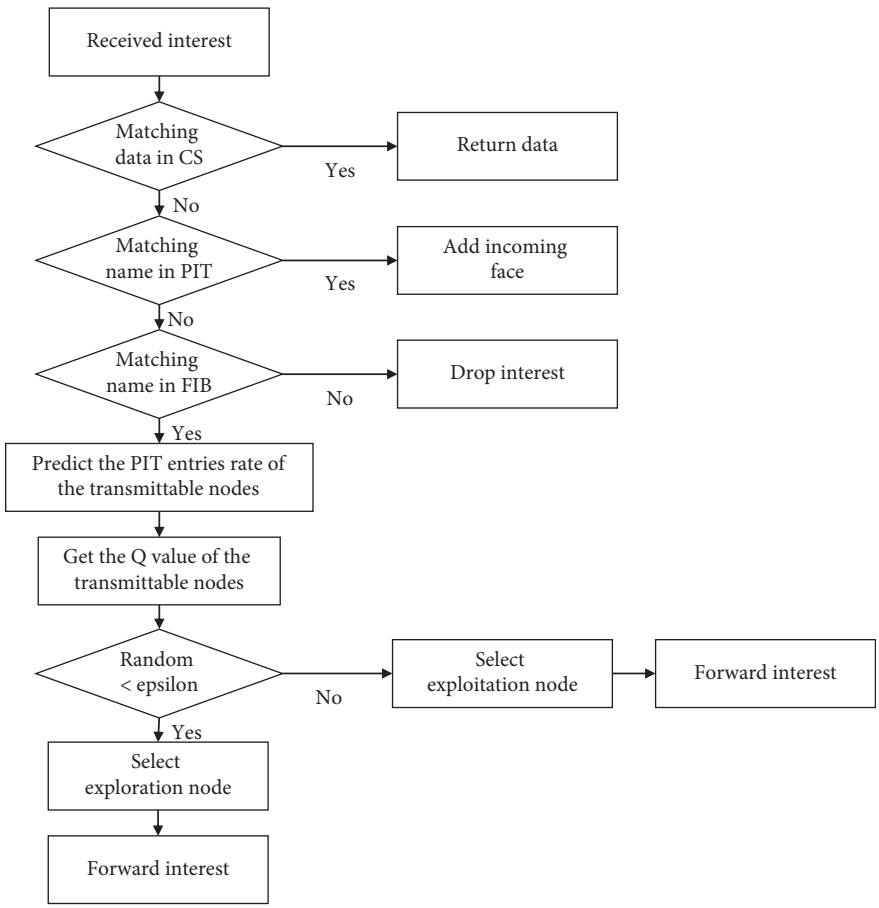


Figure 4: Q-learning forwarding strategy when the NDN node receives the interest packet.

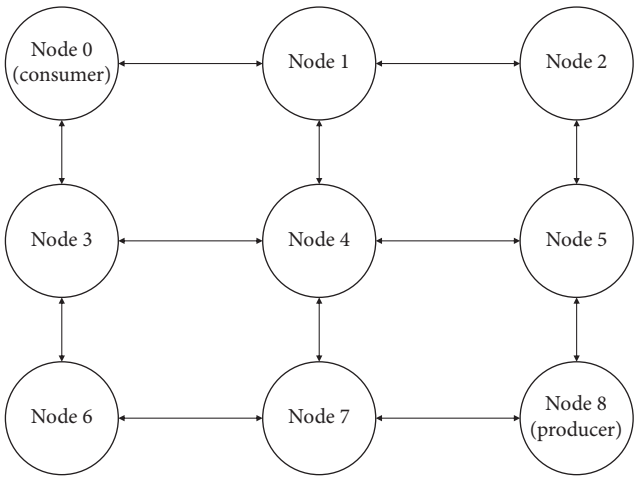


Figure 5: Topology used in the experiment.

4.2. Performance Analysis

4.2.1. Low-Level Congestion. We designed a 3 x 3 grid topology as shown in Figure 5. N5-N8 in Figure 6(a), N1-N2 and N4-N5 in Figure 6(b), and N1-N4, N4-N5, and N5-N8 in Figure 6(c) have a bandwidth of 1 Mbps, while the rest of the link bandwidth was connected at 10 Mbps. The link delay is commonly set to 10 ms. Therefore, as shown in Figures 6(a)–6(c), there are paths without bottleneck links: N0-N3-N6-N7-N8.

The graph in Figure 7 shows the average of the data packets received per second from the consumer in the three cases of Figures 6(a)–6(c). The IFS-QLSTM showed almost similar performance to that of ASF and a 17.3% higher data receiving rate than the BestRoute. The graph in Figure 8 is the average of the total packet drops in Figures 6(a)–6(c). Since there are 35,750 packets transmitted, ASF, BestRoute, and the IFS-QLSTM show packet drop rates of 0.07%, 15.9%, and 0.09%, respectively. Like the data receiving rate, the packet drop rate is similar to ASF and is 15.81% lower than BestRoute.

In detail, looking at the data rates in Figures 6(a)–6(c), you can see how each method transmits the packet. In the case of ASF, the SRTT of the adjacent nodes is measured periodically, so it quickly detects bottleneck links, finds alternate links, and sends packets to show a high InData rate. In the case of BestRoute, an alternative route is selected only when the FIB is updated, but because the update is not performed frequently or is not performed at the optimal time, packets are transmitted through the bottleneck link to show a low InData rate. Finally, the proposed method has a slightly lower initial InData rate because it transmits even paths with a low Q value due to exploration at the beginning. However, through reward, the model trains the PIT entry rate that does not cause the packet drop and the appropriate

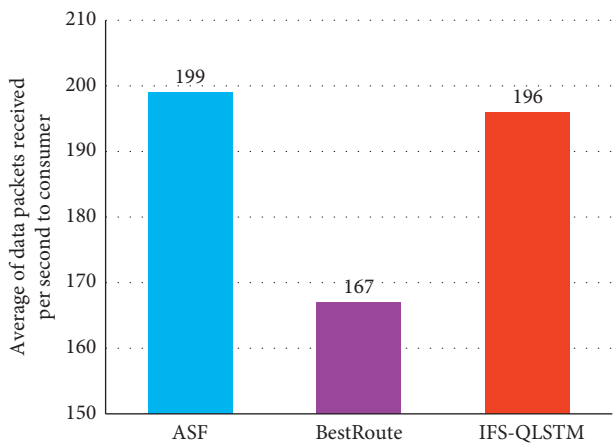


Figure 6: Comparison of the InData rate and the packet drop rate for each node. (a) Bottleneck link: N5-N8. (b) Bottleneck link: N1-N2 and N4-N5. (c) Bottleneck link: N1-N4, N4-N5, and N5-N8.

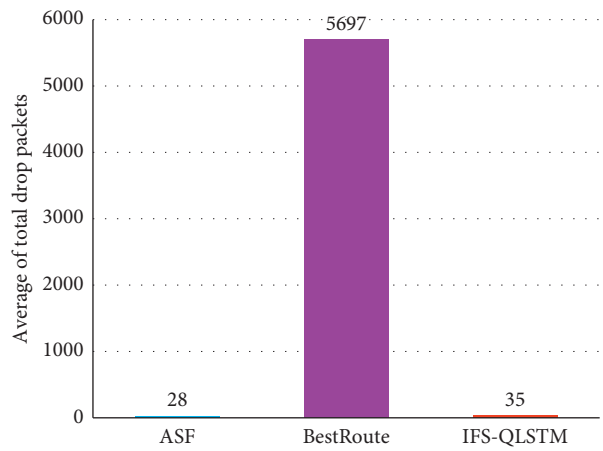
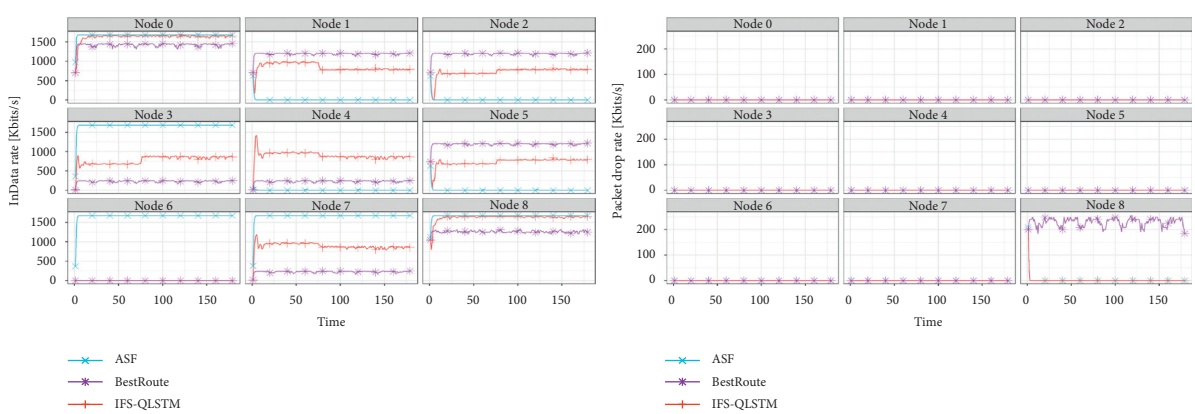


Figure 7: Comparison of the average received data rate between IFS-QLSTM and ASF and BestRoute.



(a)

Figure 8: Continued.

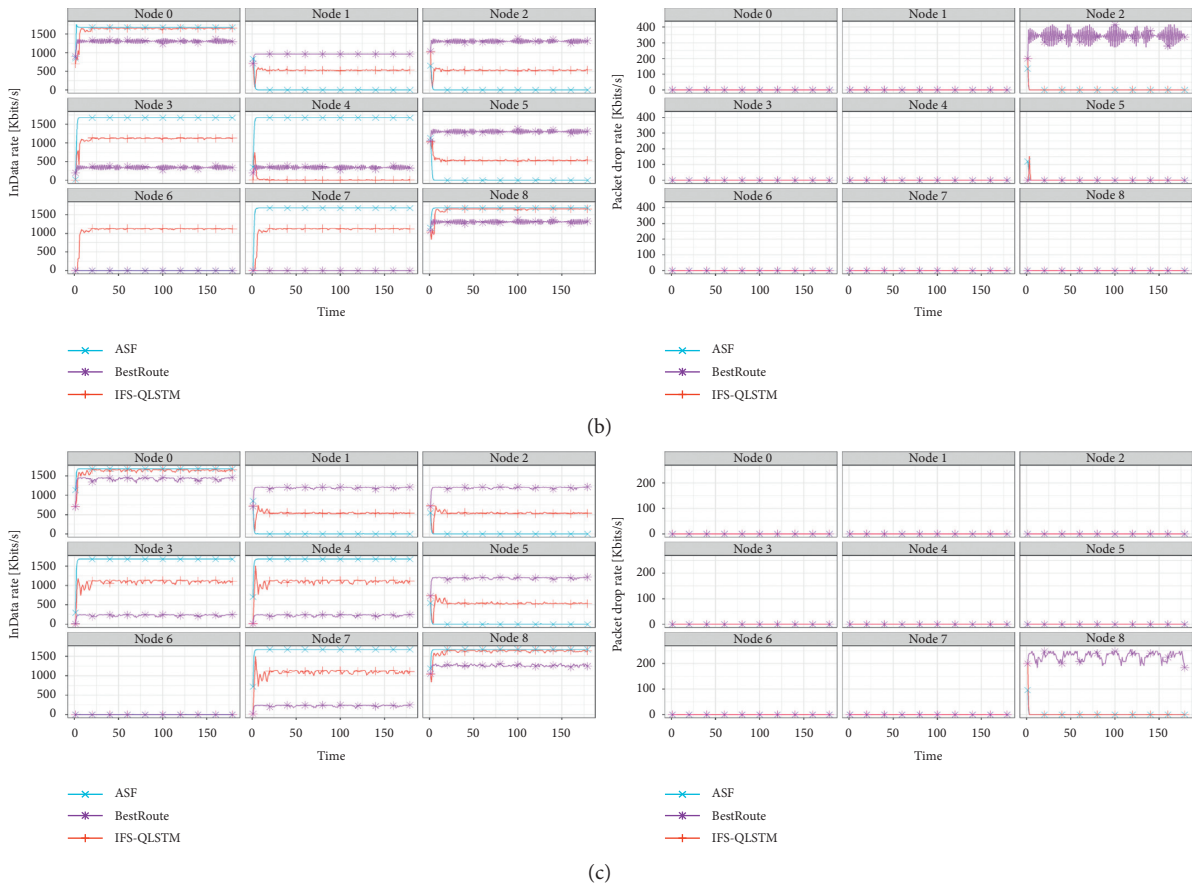


Figure 8: Comparison of the average packet drop between IFS-QLSTM and ASF and BestRoute.

amount of transmission according to the PIT entry rate for each node. Through this, the packet is properly divided into a bottleneck path and an alternate path and transmitted. Therefore, it shows an InData rate similar to ASF. In addition, looking at the packet drop rate in Figure 6, BestRoute cannot find an alternative path, resulting in high packet drops on the bottleneck link. On the other hand, in the ASF and IFS-QLSTM, a packet drop occurs briefly at the beginning, and a packet drop does not occur after finding an alternative path.

4.2.2. High-Level Congestion. We designed a 3x3 grid topology as shown in Figure 5. N5-N8 and N7-N8 in Figure 9(a), N1-N2, N4-N5, and N7-N8 in Figure 9(b), and N1-N2, N3-N6, N4-N5, and N4-N7 in Figure 9(c) have a bandwidth of 1 Mbps, while the rest of the link bandwidth was connected at 10 Mbps. The link delay is commonly set to 10 ms. Therefore, as shown in Figures 9(a)–9(c), the bottleneck links exist no matter which path from the consumer to producer is selected.

The graph in Figure 10 shows the average of the data packets received per second from the consumer in the three cases of Figures 9(a)–9(c). IFS-QLSTM showed 15.3% and 21.1% higher data rates than ASF and BestRoute. The graph in Figure 11 is the average of the total packet drops in Figures 9(a)–9(c). Since there are 35,750 packets transmitted, ASF, BestRoute, and the IFS-QLSTM show packet drop

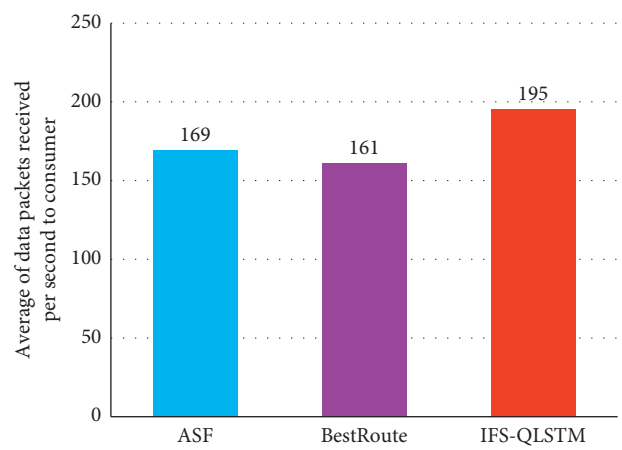


Figure 9: Comparison of the InData rate and the packet drop rate for each node. (a) Bottleneck link: N5-N8 and N7-N8. (b) Bottleneck link: N1-N2, N4-N5, and N7-N8. (c) Bottleneck link: N1-N2, N3-N6, N4-N5, and N4-N7.

rates of 14.7%, 18.8%, and 0.16%, respectively. In the case of this experiment, IFS-QLSTM shows overall higher performance than ASF and BestRoute.

In detail, by looking at the InData rate and the packet drop rate in Figures 9(a)–9(c), you can see how each method transmitted the packet and where it was dropped. In the case

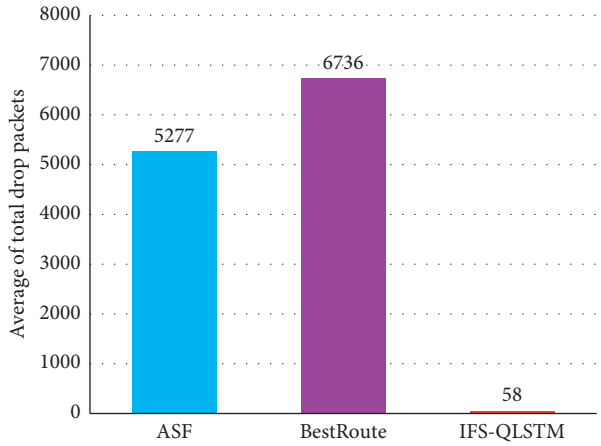
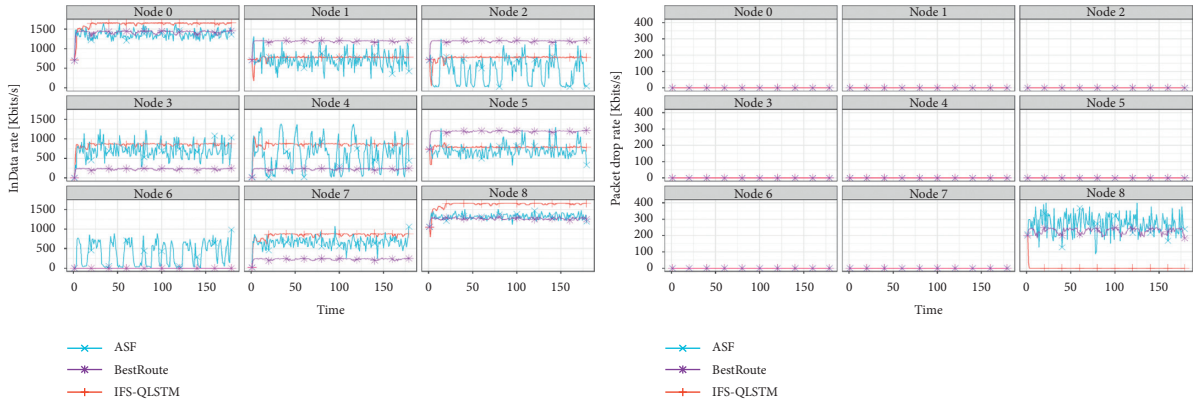
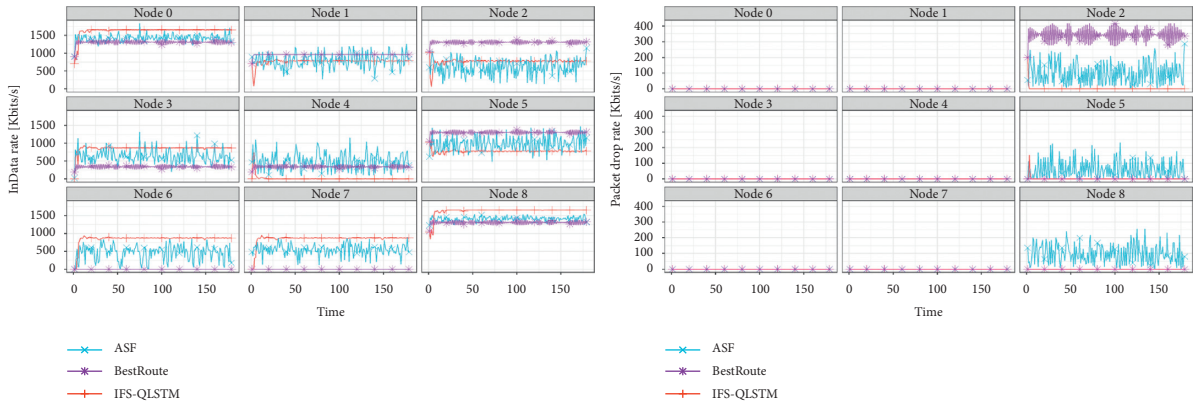


Figure 10: Comparison of the average received data rate between IFS-QLSTM and ASF and BestRoute.

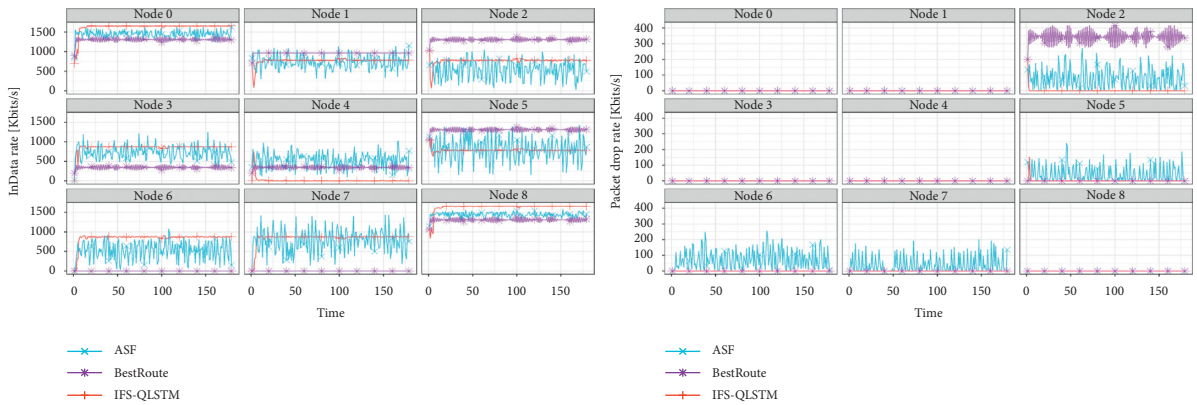


(a)



(b)

Figure 11: Continued.



(c)

Figure 11: Comparison of the average packet drop between IFS-QLSTM and ASF and BestRoute.

of ASF, unlike previous cases, it shows poor performance. The reason is that if the adjacent nodes have the same SRTT, the path is not updated in time, and thus packets are transmitted over the bottleneck link. Therefore, it shows a low InData rate. Unlike the previous case, many packet drop rates occur in the bottleneck link because the alternative path cannot be found properly. In the case of BestRoute, as before, due to the slow FIB update, a low InData rate and a high packet drop rate are shown. In the case of IFS-QLSTM, as described above, since an alternative path is selected and transmitted according to the PIT entry rate of the neighboring node, the stable packet transmission is shown even in the bottleneck link. Therefore, by achieving a high InData rate and low packet drop rate, we prove that the performance is more effective than those of ASF and BestRoute.

5. Conclusions

In this paper, we propose IFS-QLSTM, an intelligent forwarding strategy for congestion control using Q-learning and LSTM in named data networking. The proposed method first trains the LSTM model using the PIT entry rate which can be used as a congestion detection indicator by knowing the amount of data to be returned in the future. After this step, Q-learning detects the congestion of the adjacent node through the PIT entry rate predicted by the trained LSTM model and forwards it to the appropriate path. As a result of the simulation, it was verified that IFS-QLSTM has a high data rate and low packet drop compared to BestRoute and ASF by selecting the bottleneck link and the alternative link well and transmitting the packet. Therefore, it is shown that the proposed method is efficient and reliable. This suggests that there is potential for it to be used as an effective congestion control algorithm for applications to which NDN will be applied in the future.

Future work will focus on evaluating our approach in various topologies and linking it with window-based congestion control algorithms. This approach will lead to improving the congestion control performance of IFS-QLSTM.

Data Availability

The data used to support the findings of this study have not been made available because this work has been supported by the Korean government and the data cannot be publicly open.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Institute for Information & Communications Technology Promotion (no. 2015-0-00816) and by the Korea University of Technology and Education (KOREATECH), Education and Research Promotion Program (2021).

References

[1] G. Xylomenos, C. Ververidis, V. Siris et al., "A survey of information-centric networking research," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, 2013.

[2] L. Zhang, D. Estrin, J. Burke et al., "Named data networking (NDN) project," *Transportation Research Record Journal of the Transportation Research Board*, vol. 1892, p. 1, 2010.

[3] L. Zhang, A. Afanasyev, J. Burke et al., "Named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, 2014.

[4] Named Data Networking Project, https://named-data.net/.

[5] O. Akinwande, "Interest forwarding in named data networking using reinforcement learning," *Sensors*, vol. 18, no. 10, p. 3354, 2018.

[6] D. Lan, X. Tan, J. Lv, Y. Jin, and J. Yang, "A deep reinforcement learning based congestion control mechanism for NDN," in *Proceedings of the 2019-2019 IEEE International Conference on Communications (ICC)*, Shanghai, China, May 2019.

[7] Z. Rezaeifar, J. Wang, H. Oh, S.-B. Lee, and J. Hur, "A reliable adaptive forwarding approach in named data networking," *Future Generation Computer Systems*, vol. 96, pp. 538–551, 2019.

[8] G. Carofiglio, M. Gallo, and L. Muscariello, "ICP: design and evaluation of an interest control protocol for content-centric networking," in *Proceeding Of IEEE INFOCOM Workshop On Emerging Design Choices in Name Oriented Networking (INFOCOM NOMEN)*, 2012.

[9] S. Salsano, A. Detti, M. Cancellieri, M. Pomposini, and N. Blefari-Melazzi, "Receiver-driven interest control protocol for content-centric networks," in *Proceeding of the ACM SIGCOMM Workshop on Information Centric Networking (ICN)*, Helsinki, Finland, August 2012.

[10] L. Saino, C. Cocora, and G. Pavlou, "CCTCP: a scalable receiver-driven congestion control protocol for content centric networking," in *Proceeding of the IEEE ICC'13*, Sydney, Australia, September 2013.

[11] N. Rozhnova and S. Fdida, "An effective hop-by-hop interest shaping mechanism for CCN communications," in *Proceeding of the IEEE INFOCOM NOMEN Workshop*, Orlando, FL, USA, March 2012.

[12] Y. Wang, N. Rozhnova, A. Narayanan, D. Oran, and I. Rhee, "An improved hop-by-hop interest shaper for congestion control in named data networking," in *Proceeding of the ICN'13*, Hong Kong, China, August 2013.

[13] A. Afanasyev, J. Shi, B. Zhang et al., *NFD Developer's Guide Technical Report NDN-0021*, Department of Computer Science, University of California, Los Angeles, CA, USA, 2014.

[14] G. Carofiglio, M. Gallo, L. Muscariello et al., "Optimal multipath congestion control and request forwarding in information-centric networks: protocol design and experimentation," *Computer Networks*, vol. 110, 2016.

[15] C. Yi, A. Afanasyev, I. Moiseenko et al., "A case for stateful forwarding plane," *Computer Communications*, vol. 36, no. 7, 2013.

[16] V. Lehman, A. Gawande, B. Zhang et al., "An experimental investigation of hyperbolic routing with a smart forwarding plane in NDN," in *Proceedings of the IEEE/ACM 24th International Symposium on Quality of Service (IWQoS)*, Passau, Germany, June 2016.

[17] S. Mastorakis, A. Afanasyev, I. Moiseenko, and L. Zhang, ndnSIM 2.0: A New Version of the NDN Simulator for NS-3 NDN, Technical Report NDN-0028, 2015.

[18] S. Mastorakis, A. Afanasyev, and L. Zhang, "On the evolution of ndnsim: an open-source simulator for ndn experimentation," *ACM SIGCOMM Computer Communication Review*, vol. 47, no. 3, 2017.