# Deep Reinforcement Learning Based Congestion Control Mechanism for SDN and NDN in Satellite Networks

Ziyang Xing[1], Hui Qi[1], Xiaoqiang Di[1,2](✉), Jinyao Liu[1], and Ligang Cong[1]

[1] School of Computer Science and Technology, Changchun University of Science and Technology, Changchun, China
dixiaoqiang@cust.edu.cn

[2] Information Center, Changchun University of Science and Technology, Changchun, China

**Abstract.** In a satellite network, the content-centric information center network can reduce redundant data and decouple the location of network entities from the content, which is especially suitable for sending massive data from satellites to the ground. Influenced by outages, the information center network congestion control is inefficient and adaptive, the congestion policy cannot be changed from a global perspective, and the paths saved in the FIB and PIT are prone to failure. This paper proposes a congestion control algorithm based on deep reinforcement learning: RL-ICN-CC, which uses a software-defined network controller to obtain the state information of the whole network, deep reinforcement learning realizes an adaptive congestion control mechanism, and consumers adjust the interest packets according to the global state of the network CWND of the sending rate to avoid congestion. In this paper, FIB and PIT are redesigned. When the saved path easily fails, consumers can still calculate other cache locations to obtain content. Compared with other algorithms in multiple scenarios, the throughput of the proposed scheme is improved by 11%, and congestion adaptability is achieved.

**Keywords:** satellite network · congestion control · deep reinforcement learning · ICN

## 1 Introduction

Low-orbit satellite networks (LEOs) have the advantages of wide coverage and low cost and have become a future network development trend. In the highly dynamic network environment of LEO, the network architecture has difficulty meeting the needs of massive data transmission, and it is difficult to distribute satellite data to the ground [1]. Therefore, content-based network architecture (ICN), which is designed to solve the problem of large-scale content distribution [2], separating content from location, retrieving data throughout the network through content identification [3], supporting multiple access, etc., was developed. The traditional end-to-end transmission mode can realize multiterminal asynchronous data transmission, which greatly improves the transmission efficiency. The ICN congestion control directly affects the success of content


R. Li et al. (Eds.): MOBILWARE 2022, LNICST 507, pp. 13–29, 2023.
https://doi.org/10.1007/978-3-031-34497-8_2

retrieval, but the ICN routers (with routing and caching functions) are distributed. However, it is very difficult to collect global information of the entire network, and the congestion control mechanism cannot be implemented globally. Congestion control is inefficient [4], and it is easy for consumers to fail to obtain content. To solve this challenge, software-defined network (SDN) technology was introduced. The SDN controller is used to centrally control the entire network, open and configurable, etc., to obtain the status information of all links and realize online learning of artificial intelligence reinforcement learning from a global perspective [5]. The controller generates punishment and reward values in real time according to the status information. Through the process of "exploration-learning-application-summary-upgrading and testing", the system determines the optimal strategy, realizes adaptive congestion control [6], and has the characteristics of high efficiency and intelligence.

In addition, the satellite network topology has the characteristics of periodic connection and disconnection. In the TCP/IP network, the server will retransmit to solve the problem of data loss. However, in the information center network, the content of the producer is stored in the cache. When a link is disconnected (there are multiple satellite links and other links are connected), the path information in the FIB and PIT will be invalid. Therefore, this paper modifies FIB and PIT and proposes FIB and PIT based on geographic location (longitude and latitude). When a satellite link fails, the predictability of satellite topology is used to find other available cache content from FIB and PIT to improve consumption. The success rate of the user hitting the cache.

The main contributions of this paper are as follows:

(1) The FIB and PIT based on geographic location in the satellite network are proposed, and fields such as latitude and longitude are newly added. When a certain path fails, other cache paths can be found.
(2) Under the software-defined information center network, the adaptive congestion control algorithm of artificial intelligence reinforcement learning online learning is proposed, and congestion control is realized according to the state information of the entire network.
(3) To verify the modified FIB and PIT and the reinforcement learning congestion control algorithm, the Iridium constellation is designed in STK to simulate different network states, and it has been proven by a large number of experiments that congestion control can be achieved.

The rest of this paper is organized as follows: Sect. 2 introduces the related work of this paper, Sect. 3 details the adaptive congestion control algorithm for artificial intelligence reinforcement learning online learning under the software-defined information center network, Sect. 4 evaluates the performance of the proposed scheme, and finally, the conclusion and the prospect of future work.

## 2 Related Work

Network congestion means that the transmission volume exceeds the maximum load capacity of the link, and it is also an important indicator that affects the transmission performance. One of the goals of studying congestion control is to adjust the congestion window according to the real-time bandwidth to minimize the communication delay and avoid the impact of link interruption data transmission. Since the information center network congestion is different from the traditional TCP/IP congestion control mechanism, it is necessary to implement congestion control according to the characteristics of the information center. In recent years, scholars at home and abroad have mainly studied NDN network congestion control from the following aspects:

Congestion control for each hop, that is, to control the queue length for each hop. Lan D et al. [7] designed a hop-by-hop congestion control method to detect the queue length of interest packets in the queue of intermediate nodes. When the queue occupancy rate is low, the router port is allowed to forward the Interest packets; otherwise, the Interest packets are directly discarded. When the delay in LEO is too large, it will affect the monitoring rate of the data packets in the monitoring queue of the router. This solution cannot be applied on a large scale in LEO.

Receiver-driven congestion control. In the literature [8], this type of congestion control mechanism is similar to TCP/IP by confirming congestion through RTT delay and then avoiding congestion by adjusting the data transmission rate of Interests by the receiver to adapt to the transmission performance of the link (also known as “Interest Shaping”). Interest packets stored by nodes in the ICN correspond to different data. When there is a data request, the data packets corresponding to the Interest packets may be returned from different nodes [9]. The different positions of the nodes have a greater impact on the calculation of the returned data RTT [10]. Furthermore, LEO The RTT caused by the simultaneous connection and disconnection of the medium link is inaccurate, and it is also impossible to accurately determine the congestion.

In addition, many scholars have applied artificial intelligence to congestion control. Xu et al. [11] designed a framework DRL-TE based on the reinforcement learning deep deterministic policy gradient (DDPG). Using the link bandwidth, the experiment shows that the utilization of the link is improved and the end-to-end delay is reduced, but the dynamic network is not compared in the experiment.

In summary, although there have been many research results on the optimization method of information center network congestion control, due to factors such as the periodic connection and disconnection of satellites and the limited processing capacity of satellites, congestion control is still unable to adapt itself. There are few schemes for congestion control under dynamic networks. Based on this, this paper uses SDN to collect overall network status information and uses punishment and reward to achieve self-adaptation under dynamic network congestion.

## 3 System Architecture

According to the iridium constellation structure, the entire system architecture is designed as shown in Fig. 1. There are producers, switches with NDN functions, consumers (multiple), controllers, ground station nodes, etc., to form a software-defined information center network, simulating the realization of iridium constellation sends data to terrestrial consumers.

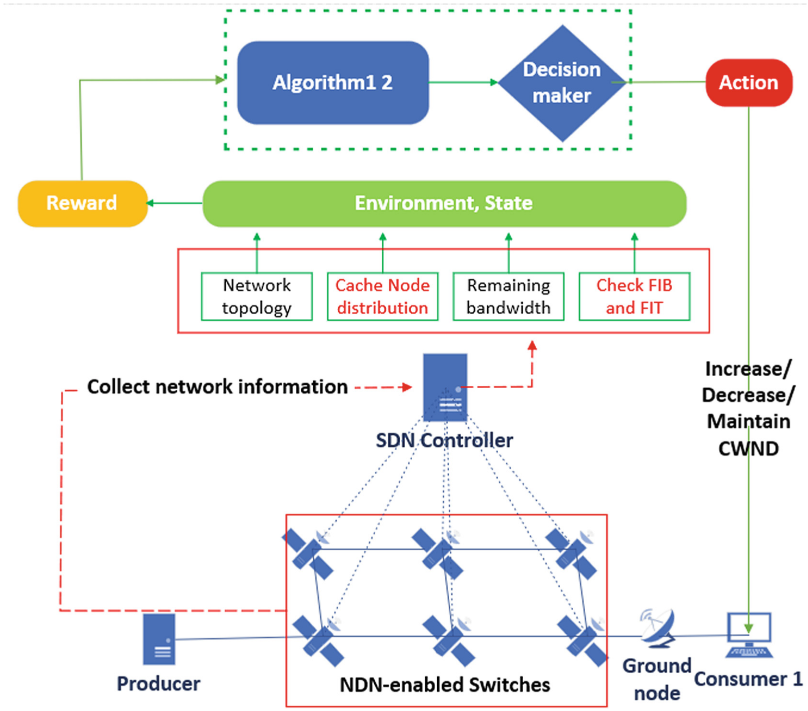



**Fig. 1.** System Architecture.

Producer: publishes content.

Consumer 1: Get some content.

Switches with NDN functions: There are 66 switches in the SDN network (see Sect. 4.1 for details), with NDN functions, such as caching and routing, and communicate with the controller through the openFlow protocol. The modified FIB and PIT are installed (see Sect. 3.2 for details).

Controller: The core of the entire network, which collects link network status, issues and installs flow tables. The controller is equipped with an artificial intelligence module, which can realize artificial intelligence online learning to achieve adaptive congestion control (see Sect. 3.3 for details).

The whole architecture realizes the content released by the producer, the storage path of FIB and PIT, the content cached, and the content obtained by the consumer. The controller collects the status information of each link, realizes the congestion control of artificial intelligence online learning, and adjusts the sending rate CWND of the consumer to achieve the purpose of increasing throughput.

## 3.1 Collect Network Status

The purpose of collecting the network status of each link is for the basis of artificial intelligence decision-making, and the status of each link is collected through the controller LLDP (Link Layer Discovery Protocol) protocol [12], mainly including the following:

Network topology: including the connection and disconnection of each link.

Cache distribution: Obtaining the cache distribution on switches with the NDN function is beneficial for consumers to obtain content.

Remaining bandwidth: This is an important parameter of congestion control, and congestion control is adjusted according to the remaining bandwidth.

Update FIB and PIT: Update cache path information, geographic location information, time, etc., in the table.

The above state information is collected to realize the parameters of artificial intelligence online learning decision-making and update them in the cycle. The artificial intelligence online learning algorithm will execute a certain action based on this to implement CWND that adjusts the consumer's sending rate.

### 3.2 Geo-Based FIB and PIT

The periodic on-off of the satellite network belongs to a dynamic network. Affected by the periodic interruption and connection of the satellite topology, the path saved in the FIB and PIT will fail, resulting in failure of consumers to obtain content. This article modifies the FIB and PIT according to the characteristics of the satellite network. Added information such as path generation time, longitude and latitude is shown in Table 1 and 2. When the consumer fails to obtain the content, the controller can retrieve all available content by looking up the FIB and FIT tables to deal with the periodic interruption and connection of the satellite, as in Algorithm 1.

**Table 1.** Modified FIB.

| key | content | cache time | position | set |
|---|---|---|---|---|
| … | | | | |
| 130 | C40 | 1663451240 | (38,4) | [W4, W6,…] |
| 140 | C471 | 1663469240 | (18,18) | [W1] |
| … | | | | |

**Algorithm 1.** Obtain the cache path algorithm.

```
input    Content unique Key
output   FIB unique Key or FIT unique Key

if FIB Content unique key ! = NULL and find with Longitude and latitude
     return Content unique Key
else if FIT unique key != NULL and find with Longitude and latitude
     return Content unique Key
else if wait for next period
     return Content unique Key in next period
else
     producer pull content again and update FIB/FIT
```

**Table 2.** Modified PIT.

| key | content | cache time | position | set |
|---|---|---|---|---|
| … | | | | |
| 13 | C436 | 1663451240 | (38,4) | [W4, W6,…] |
| 14 | C472 | 1663469240 | (18,18) | [W1] |
| … | | | | |

key: unique key.
content: content unique Key.
cache time: cache created timestamp.
position: longitude and latitude.
set: caching node set.

Lines 05–06: the consumer waits for the next cycle to obtain data again; Line 08: none of the other solutions are successful, and the producer pushes the content to the cache and updates the FIT, FIB and other information. According to Algorithm 1, due to the FIB With path invalidation in PIT, look for available cache content in the 3 selected tracks (see Fig. 2).

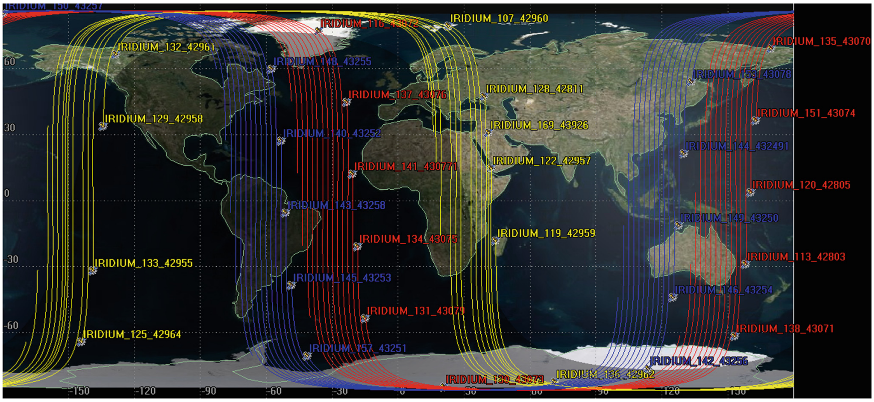



**Fig. 2.** 2D image with 3 tracks (yellow, red, blue) selected. (Color figure online)

### 3.3 Online Learning of Congestion Control

Congestion control online learning is the core of the entire architecture. As shown in Fig. 3, congestion control online learning applies deep reinforcement learning to generate real-time optimal solutions according to the network state information collected in the network state collection module. First, the agent extracts features through the convolution layer of the neural network according to the previously obtained state and then maps it into the probability of a certain action by the fully connected layer. To reuse

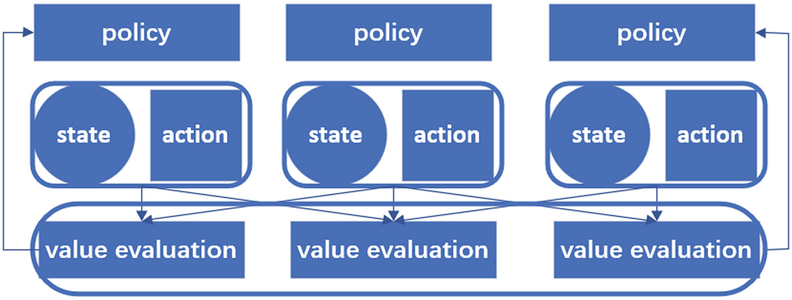



**Fig. 3.** MADDPG Congestion Control Online Learning.

the previous important experience, the experience playback mechanism is introduced, which can avoid related sequences. The samples affect the training and finally adjust the consumption sending rate CWND according to Algorithm 2.

The multiagent algorithm implements a DDPG algorithm for each agent, and all agents share a centralized critic network.

**State:** Define state S as composed of the following components: available link bandwidth, $S = (remainingbandwidth)$, obtained by the SDN controller.

**Action:** Define action A to adjust the consumer's sending rate. There are the following actions (see Table 3).

**Table 3.** Action execution types

| Way of changes | Extent of Change (CWND) |
|---|---|
| Increase | +LAV, * LAV, * 2 * 1 LAV |
| Decrease | − LAV, * 0.75 * LAV, * 0.1* LAV |
| Maintain | 0 |

LAV (last average value) = last time average value

They are the average value of CWND in the previous stage of the algorithm. When the action is "Increase", it is the average value of CWND in the previous stage, multiplied by the average value of CWND in the previous stage, and multiplied by 2 times the previous stage. Average value of CWND.

**Reward:** The reward function is to reward those that are beneficial to alleviating congestion and punish the other way around. as follows:

$$R = \alpha \times \log(throughput(t)) - \beta \times \log(RTT(t)) \quad (1)$$

α and β are parameters between [0, 1], throughput(t) is the throughput at time t, and RTT(t) is the throughput at time t.

**Algorithm 2**. MADDPG Online Learning Congestion Algorithm.

input Network topology G
output null

Randomly initialize **Actor** and **Critic** of each agent
for period $e = 1 - E$ do
  Initialize a random process N for action exploration
  Get the initial values of all agents$x$
  for $t = 1 \rightarrow T$ do:
    for each agent $i$, select an action with the current policy$a = N + \mu(o)$
      execute action $a$ and get rewards and new observations$x'$
      Randomly select some data from$D$
      for each agent, the centralized training **Critic** network
      for each agent, the centralized training **Actor** network
      for each agent, update target **Actor** network and target **Critic** network
  end for
end for

The gradient of a deterministic policy is:

$$\nabla_{\theta} J \approx \frac{1}{N} \sum_{i=1}^{N} \nabla_{\theta} u_{\theta}(s_i) \nabla_a Q_{\omega}(s_i, a)|_{a=\mu_{\theta} a(s_i)} \tag{2}$$

Each agent is a separate DDPG algorithm that then integrates the results of each agent and finally outputs the conclusion after comprehensive processing.

## 4 Simulation Test and Performance Evaluation

This section compares the comprehensive performance of the proposed scheme, such as throughput, by simulating the satellite network environment.

Computer: Intel(R) Core(TM) i5 12400F CPU @2.50 GHz × 6 processor, 32 G memory. Operating System: Ubuntu 22.10.

Controller: Ryu [13], deployed in Geostationary Earth Orbit (GEO), the operation period is equal to the rotation period of the earth, and it is stationary relative to the ground station and can obtain satellite status information in LEO.

Analysis tool: Wiresharkv3.6.3 version, used for network protocol analysis, routing analysis, packet analysis, etc. [14].

Network traffic simulation tool: Manimahi is used to simulate network parameters in simulation experiments and can record and playback operations [15].

The ground station is located in Jiamusi, China, in the Northern Hemisphere.

### 4.1 Simulation Test

$$X' = \begin{bmatrix} 1 & X_{(1,2)} & X_{(1,3)} & \ldots & X_{(1,k)} \\ X_{(2,1)} & 1 & X_{(2,3)} & \ldots & X_{(2,k)} \\ X_{(3,1)} & X_{(3,2)} & 1 & \ldots & X_{(3,k)} \\ \ldots & \ldots & \ldots & 1 & \ldots \\ X_{(k,1)} & X_{(k,2)} & X_{(k,3)} & \ldots & 1 \end{bmatrix} \tag{3}$$

The position information of each satellite derived from STK is shown in formula (6), which represents the position matrix of each satellite in the iridium constellation. $X_{(j,k)}$. Since the iridium constellation switches every 60 s, it runs in sequence. When a satellite is connected to multiple satellites, this article ignores the link situation that is too far from the satellite, only discusses the one-to-one corresponding satellite link, and organizes the satellite distance matrix $X$' into the visibility matrix $X$, as follows.

$$X = \begin{bmatrix} 0 & X_{(1,2)} & X_{(1,3)} & \ldots & X_{(1,k)} \\ X_{(2,1)} & 0 & X_{(2,3)} & \ldots & X_{(2,k)} \\ X_{(3,1)} & X_{(3,2)} & 0 & \ldots & X_{(3,k)} \\ \ldots & \ldots & \ldots & 0 & \ldots \\ X_{(k,1)} & X_{(k,2)} & X_{(k,3)} & \ldots & 0 \end{bmatrix} \tag{4}$$

$X$ is the satellite visibility matrix composed of 0 and 1 after being sorted by $X$'. In the $X$ matrix, except for the value with too long a distance, when there is a link between satellites, it is recorded as 1; otherwise, it is recorded as 0, as shown in formula (7). The periodicity between satellites in the iridium constellation will follow the relationship of (7). See Algorithm 3 for the pseudo code of connection and disconnection, dynamic switching link.

**Algorithm 3.** Iridium Constellation Dynamic Network Algorithm.

| | |
|---|---|
| input | matrix$X(X, X^2, \ldots, X^N)$, $N = 66$ |
| | $\Delta t = 60$ |
| | for each $X^n$ != null and $X^n \in X$ do |
| | if $X^n = X^{\Delta t+n}$ |
| | do nothing |
| | else |
| | if $X^n_{(j,k)}$ ! $= X^{\Delta t+n}_{(j,k)}$ and$X^{\Delta t+n}_{(j,k)} = 1$ |
| | put link $X^n_{(j,k)}$ $X^{\Delta t+n}_{(j,k)}$ up |
| | else if $X^n_{(j,k)}$ ! $= X^{\Delta t+n}_{(j,k)}$ and$X^{\Delta t+n}_{(j,k)} = 0$ |
| | put link $X^n_{(j,k)}$ $X^{\Delta t+n}_{(j,k)}$ down |
| | end if |
| | end for |

In Algorithm 3, lines 02–03 indicate that one cycle has been run and its state remains unchanged. $X^{n}_{(j,k)}$ represents the value of the $j$ row and $k$ column element in the $n$ visible matrix, the 05–06th row, when its value and the value of the element in the next cycle are 1, the chain The link is connected; in lines 07–08, when the value of $X^{\Delta t+n}_{(j,k)}$ is 0, the link is disconnected, and $\Delta t$ is the satellite switching time interval, and the whole process simulates the satellite Periodic on and off.

## 4.2 Performance Evaluation

The experiments in this paper are implemented on the Mini-NDN open source software [16]. This software is a network simulation platform based on Mininet, which can realize all the functions of NDN and build a software-defined information center network together with the controller and switches with NDN functions. Consumers obtain the cached content through the software-defined information center network. According to Algorithm 3 and a switch with 66 NDN functions, the periodic disconnection and connection of satellites are simulated. The comprehensive performance of the algorithm in this paper will be compared through the following experiments. To eliminate the influence of other factors on the experimental results, the average of 20 experiments will be taken into the statistical results.

(1) **Can it hit the cache**

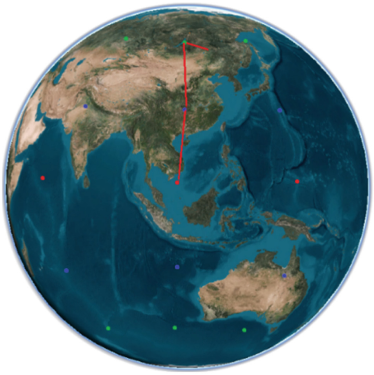
Due to the dynamic characteristics of low-orbit satellite networks (periodic connection and disconnection of intersatellite links), the path information cached in the FIB and PIT tables will become invalid; that is, consumers cannot obtain CS through the paths in the FIB and PIT content above. As shown in Fig. 4.

In Fig. 4(a), the consumer has been unable to obtain the content in the FIB and PIT tables and finally failed to obtain the content.

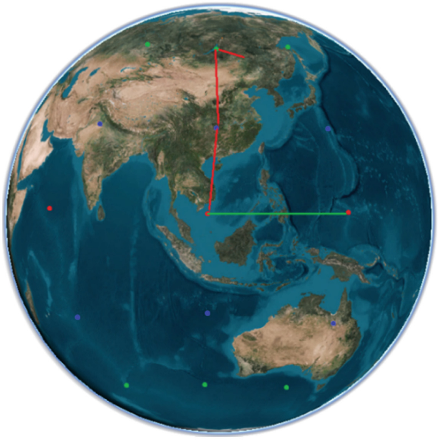
In Fig. 4(b), the consumer fails to obtain the content from the CS through the path in the FIB and PIT, but through the recalculation of the geographic location of the scheme in this paper, the routing data have been obtained from other caches, and the content is finally obtained successfully.

In Fig. 4(c), the consumer fails to obtain the content through the FIB and PIT. Through the recalculation of the geographical location in this paper, the producer provides the content again. Finally, the consumer succeeds in obtaining the content and updates the FIB and PIT.

In Fig. 4(d), the consumer fails to obtain the content through the FIB and PIT. According to the predictability of the satellite topology, the consumer waits again for the next cycle and reacquires the content according to the path in the FIB and PIT table; finally, the consumer obtains the content. Content succeeded.

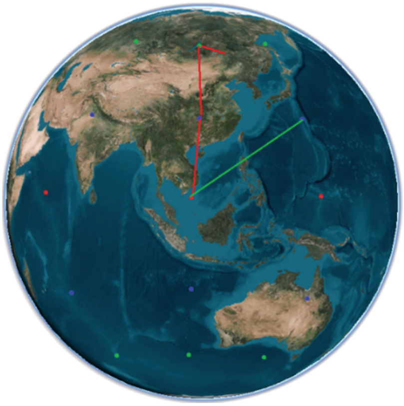


(a) The network link is interrupted, and the consumer fails to obtain the content

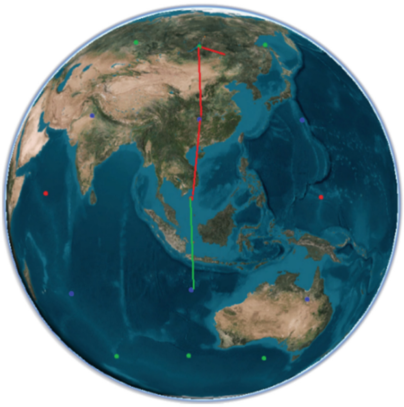


(b) Recalculate the available cache, and the consumer obtains the content successfully

**Fig. 4.** Several situations in which consumers acquire content.



(c) The available cache is recalculated, and the consumer obtains the content successfully



(d) The consumer waits for the next cycle and obtains the content successfully

**Fig. 4.** (*continued*)

### 4.3 Throughput CDF

Figure 5 Comparison of CDF between the algorithm in this paper [17] and the DPCCP algorithm the literature proposes a no artificial network utility maximization (network utility maximization) model DPCCP (delay-based path-specified congestion control protocol) to solve congestion control. In this paper, this algorithm is applied to the network of simulation 5.1 and the algorithm in this paper. About throughput The CDF is shown in Figure x. The DPCCP algorithm selects the transmission path by calculating the performance of each path, but the characteristics of periodic connection and disconnection of satellites result in a maximum throughput of 16 Mb/s, which is lower than that of RL-ICN-CC in this paper. In addition, the algorithm cannot be self-adapted to congestion control, so the CDF of the DPCCP algorithm fluctuates greatly, while the throughput

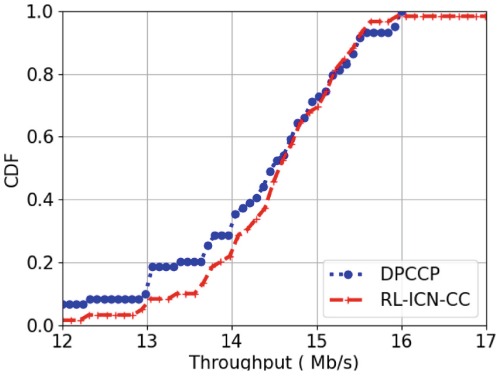



**Fig. 5.** CDF comparison between this algorithm and DPCCP algorithm.

of the algorithm in this paper is stable and does not fluctuate greatly. Reinforcement learning online learning can adapt to consumers in real time according to the network environment. Rate CWND to avoid congestion and thus improve throughput. The maximum throughput and average throughput of the RL-ICN-CC algorithm in this paper are higher than those of the DPCCP algorithm.

(2) **Buffer occupancy rate**

In this paper, the RL-ICN-CC algorithm adjusts the CWND according to the consumer's sending rate, so the occupancy of the buffer is extremely important; that is, the queue length is controlled to achieve stability. According to the algorithm (active queue management) and the experimental method in the literature [19], the number of data packets in the NDN flow is counted to analyze the buffer occupancy, as shown in Fig. 6.

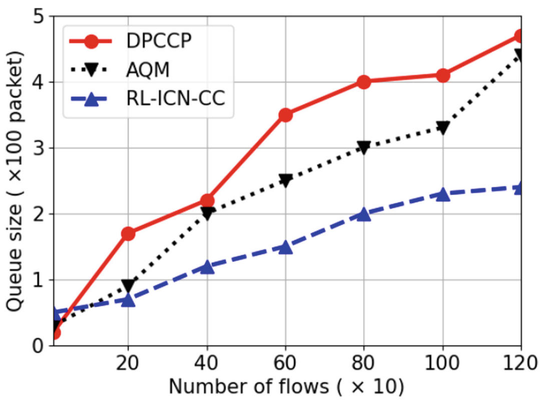



**Fig. 6.** Average queue length under different NDN flows for each algorithm.

With the increase in NDN flow, the average queue length in each algorithm increases, and the increase in DPCCP and AQM is significantly larger than that of RL-ICN-CC in this paper. This is because the other two algorithms have a weaker ability of consumers to adjust CWND. The congestion cannot be adapted well, which affects the transmission performance. With the increase in time, it is very easy for the buffer to be full and the data to overflow, and finally, the transmission fails. The RL-ICN-CC algorithm in this paper can adjust the real-time CWND and the buffer queue length without fluctuation and can cope with the characteristics of the transmission instability caused by the intermittent topology in the satellite network.

(3) **Size of PITs**

Since this paper remodifies PIT and FIB, its performance needs to be evaluated. The test counts the number of PIT packets over a period of time, and the difference in the number of PITs can reflect whether the consumer's sending rate is stable, that is, the pros and cons of congestion control.

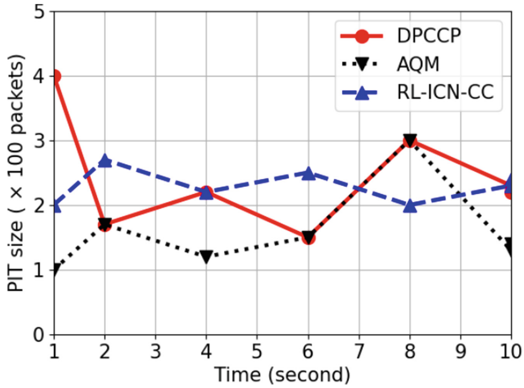



**Fig. 7.** Comparison of the number of PITs in each algorithm.

It can be seen from Fig. 7 that for the two algorithms except the RL-ICN-CC algorithm in this paper, the difference between the highest value and the lowest value is large, which reflects poor congestion control, while the RL-ICN-CC algorithm in this paper has small fluctuations and is stable at 2.2 The difference between the highest value and the lowest value is approximately 0.5, and the transmission is stable.

(4) **Content delivery rate**

The content delivery rate refers to the ratio of the amount of content obtained per unit time to the time when consumers initiate a request, which can reflect the speed of obtaining the content. or unable to obtain content.

As shown in Fig. 8, the content delivery rate of each algorithm within 32 s is displayed through a scatter plot. The algorithm RL-ICN-CC in this paper has remodified PIT and FIB. In the satellite network, the efficiency of content acquisition is significantly improved, and it can obtain content in a short time. DPCCP cannot cope with the drawbacks of periodic connection and disconnection of satellite network topology. All the time required is long, and sometimes the content cannot be obtained due to link interruption.

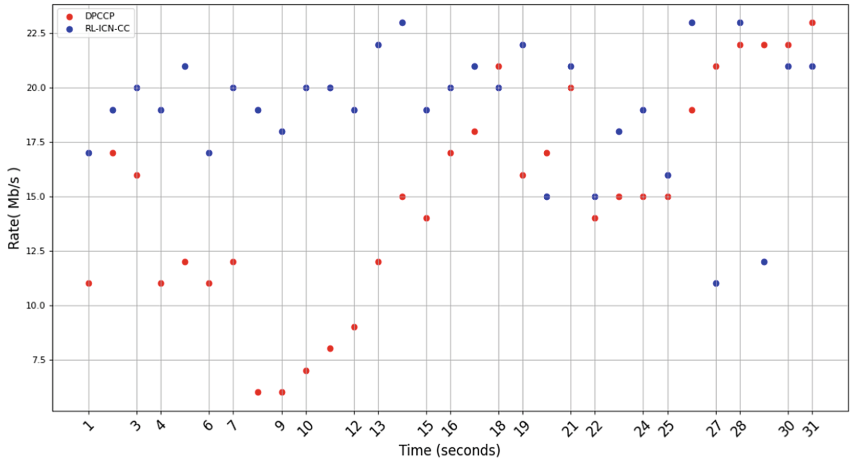



**Fig. 8.** Content Delivery Rate Comparison.

Furthermore, the average rate of the algorithm RL-ICN-CC is significantly higher than that of DPCCP at 11 Mb/s at 19 Mb/s, and the algorithm RL-ICN-CC is relatively stable and has less fluctuation, which can continuously transmit data and is suitable for the batch distribution of large files.

(5) **End-to-end delay**

The end-to-end delay is an important indicator of whether the entire link is congested or not and the performance of the link. The same link and content were selected in the experiment to compare the advantages and disadvantages of the algorithm in this paper and other algorithm.

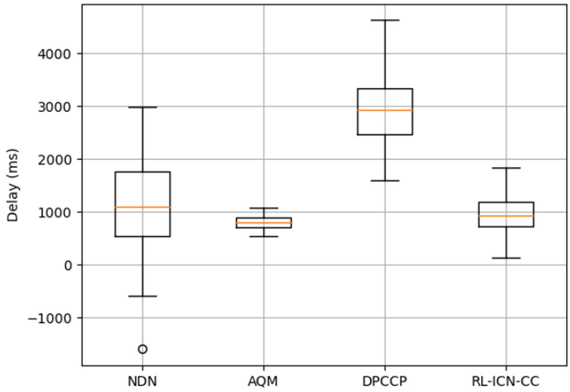



**Fig. 9.** Delay box diagram of each algorithm

As shown in Fig. 9, when the reinforcement learning NDN is not used, the delay is large, and the difference between the maximum value and the minimum value is also large. The RL-ICN-CC algorithm in this paper realizes adaptive congestion control, and the overall performance is better than that of the other algorithms.

## 5 Conclusion and Future Work

In this paper, a content-centric information center network is proposed to solve the problem of satellite mass data delivery to the ground. However, due to the dynamic nature of satellite topology, the network is prone to congestion, and user and cache paths are prone to failure. It is proposed to use the SDN controller to obtain the status information of the entire network. Reinforcement learning solves the optimal control strategy, avoids congestion by adjusting the CWND of the consumer interest packet sending rate, redesigns the PIT and FIB including the latitude and longitude, and refinds the content of other cache servers according to the characteristics of topology predictability, avoiding consumers. The content acquisition fails due to path interruption. Through simulation experiments, it is known that the proposed scheme has strong adaptability and generalization ability in the satellite network and improves the throughput. In future work, other deep reinforcement learning algorithms will be further studied to solve the problem of avoiding congestion in information-centric networks.

**Acknowledgments.** This research was funded by the National Natural Science Foundation of China under grant No. U21A20451, the Science and Technology Planning Project of Jilin Province under grant No. 20200401105GX and 20220101143JC, and the China University Industry-Academia-Research Innovation Fund under grant No. 2021FNA01003.

## References

1. Li, J., Xue, K., Liu, J., et al.: An ICN/SDN-based network architecture and efficient content retrieval for future satellite-terrestrial integrated networks. IEEE Netw. **34**(1), 188–195 (2019)
2. Gupta, D., Rani, S., Ahmed, S.H., et al.: ICN-based enhanced cooperative caching for multimedia streaming in resource constrained vehicular environment. IEEE Trans. Intell. Transp. Syst. **22**(7), 4588–4600 (2021)
3. Leal, G.M., Zacarias, I., Stocchero, J.M., et al.: Empowering command and control through a combination of information-centric networking and software defined networking. IEEE Commun. Mag. **57**(8), 48–55 (2019)
4. Benedetti, P., Piro, G., Grieco, L.A.: A softwarized and mec-enabled protocol architecture supporting consumer mobility in information-centric networks. Comput. Netw. **188**, 107867 (2021)
5. Jmal, R., Fourati, L.C.: Content-centric networking management based on software defined networks: survey. IEEE Trans. Netw. Serv. Manag. **14**(4), 1128–1142 (2017)
6. Liu, Z., Zhu, J., Pan, C., et al.: Satellite network architecture design based on SDN and ICN technology. In: 2018 8th International Conference on Electronics Information and Emergency Communication (ICEIEC), pp. 124–131. IEEE (2018)
7. Lan, D., Tan, X., Lv, J., et al.: A deep reinforcement learning based congestion control mechanism for NDN. In: ICC 2019–2019 IEEE International Conference on Communications (ICC), pp. 1–7. IEEE (2019)

8. Carofiglio, G., Gallo, M., Muscariello, L.: ICP: design and evaluation of an interest control protocol for content-centric networking. In: 2012 Proceedings IEEE INFOCOM Workshops, pp. 304–309. IEEE (2012)
9. Ryu, S., Joe, I., Kim, W.T.: Intelligent forwarding strategy for congestion control using Q-learning and LSTM in named data networking. Mob. Inf. Syst. **2021** (2021)
10. Li, L., Xu, K., Li, T., et al.: A measurement study on multipath TCP with multiple cellular carriers on high speed rails. In: Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication, pp. 161–175 (2018)
11. Xu, Z., Tang, J., Meng, J., et al.: Experience-driven networking: a deep reinforcement learning based approach. In: IEEE INFOCOM 2018-IEEE Conference on Computer Communications, pp. 1871–1879. IEEE (2018)
12. https://learningnetwork.cisco.com/s/article/link-layer-discovery-protocol-lldp-x
13. https://ryu-sdn.org/
14. https://www.wireshark.org
15. http://mahimahi.mit.edu/
16. https://minindn.memphis.edu/
17. Ye, Y., et al.: Delay-based network utility maximization modeling for congestion control in named data networking. IEEE/ACM Trans. Netw. **29**(5), 2184–2197 (2021)
18. Xu, Y., Ni, H., Zhu, X.: An effective transmission scheme based on early congestion detection for information-centric network. Electronics **10**(18), 2205 (2021). https://doi.org/10.3390/electronics10182205
19. Li, W., et al.: Charging on the route: an online pricing gateway congestion control for ICNs. IEEE Trans. Netw. Serv. Manag. **17**(1), 239–250 (2019)