A Deep Reinforcement Learning Based Congestion Control Mechanism for NDN

Conference Paper · May 2019

DOI: 10.1109/ICC.2019.8761737

5 authors, including:

Xiaobin Tan
University of Science and Technology of China
94 PUBLICATIONS **552** CITATIONS
SEE PROFILE

Jian Yang
University of Science and Technology of China
121 PUBLICATIONS **1,190** CITATIONS
SEE PROFILE

A Deep Reinforcement Learning based Congestion Control Mechanism for NDN

Dehao Lan, Xiaobin Tan, Jinyang Lv, Yang Jin, Jian Yang
Laboratory for Future Networks
University of Science and Technology of China
Hefei, China
Emails: landehao@mail.ustc.edu.cn, xbtan@ustc.edu.cn, jinyangl@mail.ustc.edu.cn,
yjinking@mail.ustc.edu.cn and jianyang@ustc.edu.cn

***Abstract*—Named Data Networking (NDN) is an emerging future network architecture that changes the network communication model from push mode to pull mode, which leads to the requirement of a new mechanism of congestion control. To fully exploit the capability of NDN, a suitable congestion control scheme must consider the characteristics of NDN, such as connectionless, in-network caching, content perceptibility, etc. In this paper, firstly, we redefine the congestion control objective for NDN, which considers requirements diversities for different contents. Then we design and develop an efficient congestion control mechanism based on deep reinforcement learning (DRL), namely DRL-based Congestion Control Protocol (DRL-CCP). DRL-CCP enables consumers to automatically learn the optimal congestion control policy from historical congestion control experience. Finally, a real-world test platform with some typical congestion control algorithms for NDN is implemented, and a series of comparative experiments are performed on this platform to verify the performance of DRL-CCP.**

***Index Terms*—Named Data Networking, Congestion control, Deep reinforcement learning**

I. Introduction

In recent years, with the rapid increase of data transmission volume in Internet, TCP/IP and its variants have suffered from unsatisfied performance [1]. Address based communication is the essential limitation of TCP/IP, and this demands renovation to the TCP/IP networks [2]. The information-centric networking (ICN) [3] concept is a common approach for future Internet which uses content based communication. As a typical ICN architecture, Named Data Networking (NDN) [4] [5] becomes a research hotspot. And congestion control mechanism for NDN is one of the key technologies need to be studied.

Some successful experience in TCP is worth following, such as controlling the traffic by a sliding window with dynamic size. However, since the entirely new architecture of NDN, we should fully consider the challenges and opportunities to design a suitable congestion control mechanism for it:

1) Communication in NDN is consumer-driven and connectionless. It means that consumers retrieve Data packets by sending corresponding Interest packets to the network [6], and they only care about what data they want rather than where it is located. Since the "pull" mode and one-Interest-one-Data mechanism, we generally control Data return rate by adjusting the Interest sending rate, this is exactly the opposite of TCP and has a certain hysteresis. And there is no exact information about the corresponding Data packet when sending out the current Interest packet, until the Data packet is received. Additionally, due to the connectionless transport, connection-based congestion control and fairness are no longer applicable, the congestion control objective in NDN should be redefined.

2) Ubiquitous in-network caching, namely the intermediate node can cache the passing Data packets to satisfy future requests. With in-network cache, a consumer may obtain Data packets from multiple sources through multiple paths, which makes it difficult to adopt retransmission timeout (RTO) into congestion control in NDN because it is designed for single-source connection in TCP [7]. Moreover, these features make it hard to accurately measure the network state.

3) Unlike TCP/IP, the transmission in NDN is content-based (named data), the information of content transmitted is available, such as name, type, size, etc. We should fully consider the content characteristics and combine them with the requirements of consumers, use different congestion control strategies for different contents to realize more reasonable congestion control objective.

The complicated and highly dynamic network state of NDN makes it hard to model, predict and control [8]. Moreover, various contents with different characteristics, distributions and different requirements need to be satisfied. Reasonably, ideal congestion control mechanism for NDN should try to make full use of related information of content information, network state and users' requirements to achieve optimal control, and have the ability to adaptively adjust congestion control strategy for dynamic scenarios.

As an experience-driven model-free approach, deep reinforcement learning (DRL) [9] can intelligently learn optimal policies directly from high-dimensional sensory inputs, is particularly suitable for congestion control in NDN. In this paper, we propose the DRL-based Congestion Control Protocol (DRL-CCP), a novel congestion control mechanism for NDN. We make the following contributions:

- We redesign a more reasonable congestion control objective for NDN which considers requirements diversities for different contents. Moreover, leverage the connectionless

This work is supported by National Nature Science Foundation of China under Grant 61673360 and ZTE Industry-Academia-Research Cooperation Funds.

978-1-5386-8088-9/19/$31.00 ©2019 IEEE

feature of NDN, it's more meaningful to schedule multiple data flows in same host for collaborative optimization.
- We propose the DRL-CCP, a highly effective and practical DRL-based mechanism of congestion control in NDN, which can intelligently generate suitable actions for adjusting the size of Interest sending window (*cwnd*) at the consumers side.
- A congestion control prototype and testbed for NDN are developed, various typical congestion control algorithms are implemented on it, and a series of comparative experiments are performed to verify the efficiency of DRL-CCP.

The rest of this paper is organized as follows: Section II discusses related work on the topic. Section III introduces the DRL-based congestion control mechanism for NDN. In section IV, we demonstrate the effectiveness of DRL-CCP. Finally Section V concludes the paper.

II. Related Work

Research on congestion control for NDN network is still in its infancy, existing works can be roughly categorized into three types [10]: receiver-based window control mechanism, hop-by-hop rate control approach and hybrid mechanism.

Receiver-based mechanism is implemented at the consumer side by controlling the Interest sending rate [11], mainly by adjusting a congestion window which controls the number of packets injected into the network [12]. As a representative round-trip time (RTT) based mechanism, ICP [13] realizes a window-based Interest flow control using TCP-like mechanism in which packets sharing identical name prefix are considered as a flow. ICP is vulnerable to unexpected large RTT packet which is treated as a loss event even though there is no packet loss occurs and will cause the Interest sending rate to be halved. However, the RTT changes irregularly due to the in-network caching of NDN. As proposed in [14], the novel BBR mechanism performs well in TCP by continuously detecting the maximum link capacity. Since communication in TCP is point-to-point, the link remains unchanged once the connection is established, but it's different in NDN. As proposed in [15], the BBR-like congestion control algorithm for NDN performs better than ICP in some scenarios, but it isn't optimal control when considering in-network caching of NDN. Since the cache diversity and measure difficulty, the measured maximum link capacity is inaccurate. The ICTP [16], CCTCP [17], etc, also basically use RTT for the estimation of congestion.

HoBHIS [18] is a typical hop-by-hop approach which detects and controls congestion at intermediate nodes by controlling the Interest sending rate via shaping the Interests, and a NACK feedback was added to HoBHIS to notify a downstream node about congestion in [19]. As well as NHBH-RCP [20] and IRNA [10], as pointed out in [10] and [21], most existing methods in this direction (hop-by-hop Interest shaping) assume known link bandwidths and Data chunk sizes, limits their effectiveness in scenarios that underlying bandwidth is variable and unknown. CS-based algorithm [6] is designed to reduce packet loss caused by bursty traffic, it's also unable to cope with the RTT variation in NDN. In addition, making changes on intermediate nodes is more complicated and will affect the performance of these nodes.

Typical hybrid mechanisms include HR-ICP [22], CHoP-CoP [23], ECN-based [11], ECP [7], EC-CUBIC [24], PCON [21], etc. They detect network congestion state at intermediate nodes and control Interest sending rate at the receiver or intermediate nodes. However, in most of them, there is no proper collaboration between the receiver-driven and the hop-by-hop mechanism, and the performance of intermediate nodes will be affected. Methods based on congestion state detection and feedback have a certain hysteresis, the best way is to let consumer have the ability of perception and prediction. In any case, the congestion control of consumer side should be met first. The information available at consumer side is more comprehensive, including the user needs, historical control information and various information of received packets, retransmitted packets or lost packets.

Recent years have witnessed several new approaches for congestion control both in TCP and NDN. PCC Allegro [1], PCC Vivace [25] for TCP and RDPCC [2] for NDN are able to adopt actions to result in high performance by continuously observing the connection between their actions and empirically performance at the sender. CCP [26] developed a system to implement complex congestion control functions by placing them outside the datapath. NN [27], is a scheme based on Neural Network to predict the packet drop event before sending out the Interest which is implemented in each router. Similarly, ACCP [28] is an adaptive congestion control protocol based on deep learning which has considered the multi-source transport in NDN, but it is unrealistic to implement deep learning on each router considering the cost and computational capabilities. By integrating a reinforcement-based Q-learning framework, QTCP [12] enables senders to gradually learn the optimal congestion control policy in an on-line manner. They elaborate on the effectiveness of a learning based approach for TCP, but it's inaccurate to represent the network by finite number of states and will cause performance degradation.

Differ from prior works, we introduce DRL into congestion control of NDN to calculate the proper size of *cwnd* at consumers side. By redesigning the congestion control objective and making full use of information perceived by consumer, DRL-CCP designed by us achieves satisfactory performance.

III. DRL-based Congestion Control Mechanism

In this section, we first introduce the design motivation (Sec. III-A). Then Sec. III-B shows the overall architecture of DRL-CCP and explains how to apply DRL into congestion control of NDN. And the key details of the design are discussed separately in Sec. III-C.

A. Motivation

As mentioned before, unlike TCP/IP, the transmission in NDN is content-based and connectionless. So it's achievable and necessary to redesign the objective of congestion control

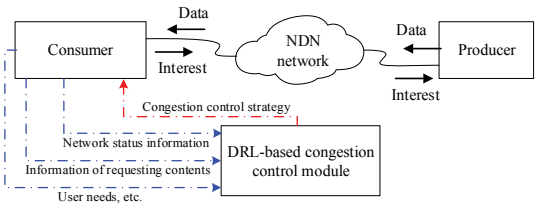


Fig. 1: Overview of DRL-CCP architecture

in NDN, take advantage of the feature that the content information is knowable to optimize the transmission in NDN networks. As introduced in Section II, traditional TCP-like congestion control mechanisms are unable to act toward high performance by learning from dynamic environments or experience, lack of perception and prediction ability. DRL combines the perception ability of deep learning with the decision-making ability of reinforcement learning, it can learn from dynamic environments and hand complicated control problems with large state space, realize directly control from raw input to output. So the DRL model is introduced for congestion control in NDN for optimization and adaptivity for dynamic environments. In addition, the receiver driven and one-Interest-one-Data transport mode enable congestion control at receiver, and to avoid impact on the performance of intermediate nodes, our mechanism is mainly deployed at consumer side (Fig. 1) where we can make full use of the computing resource of the consumer, obtain comprehensive information from the network or users and synthetically control the transmission of Interest packets from different applications. Of course, in the case that the intermediate node has relatively sufficient resources, it can be used for auxiliary information collection or control.

B. Design Overview

The framework of DRL based congestion control mechanism for NDN is shown in Fig. 2. Complicated computation of training the neural network is one of the key challenges of applying deep learning algorithm into congestion control. So we separate the training of neural network (DRL section) from the realtime data transmission (Data request section), combine offline pre-training with online selective training and updating.

In detail, we first design the integrated congestion control objective of consumer according to the user requirements, which include the characteristics of required contents, user's requirements of experience quality, etc. Next, we determine the specific DRL training model and related variables and parameters. Then we use the available data to pre-train the neural network of DRL model in the direction of approaching the integrated objective, the training data comes from the database which is used to store the data sampled from the network. In the transmission phase, the consumer sends Interest packets through the pipeline, which is designed to control the sending rate. The monitored status of the network environment determines DRL model calling strategy and parameter update strategy, the former adjusts the frequency of calling the neural networks relies on information such as packet delay and computing power of the consumer, the latter selectively updates the stored neural networks according to the network states.

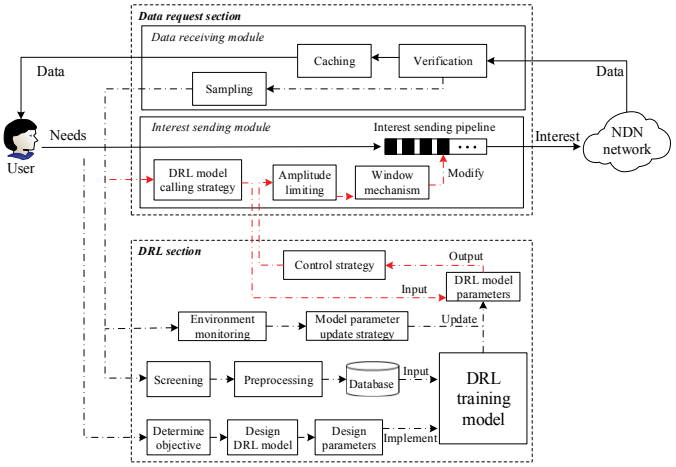


Fig. 2: Implementation of DRL-CCP mechanism

C. DRL-based Congestion Control

Before adopting DRL into congestion control, we should determine the control objective and model the congestion control as a Markov decision process (MDP), which is defined as a tuple (S, A, P, R, γ). Where S is a set of states, A is a set of actions, P is the state transition probability matrix, R is the reward function and γ is a discount factor. Since it is impossible to obtain the P in this complex environment, the reinforcement learning is used. The selection and design of above mentioned elements will be described in detail below.

*1) **Congestion Control Objective**:* Generally, traditional TCP-like congestion control mechanisms treat all the flows the same, namely same strategy is used for each flow, and in the congestion avoidance phase, multiple flows share the same network bandwidth by transmitting data at the same speed. But in NDN, optimal control can be achieved by making full use of the new features and collaborative optimization. By combining content features with user needs and special requirements of the consumer, different strategies can be used for various contents. Especially when different contents sharing a link, the content-based fairness (different transmission speeds can be used for different contents to better meet various needs) which is more reasonable for NDN can be realized. In this way, NDN networks can provide users with better service. So the substantial objective of congestion control in NDN is to provide users with best possible service while maintaining the stability of entire network. Namely, is to allow various

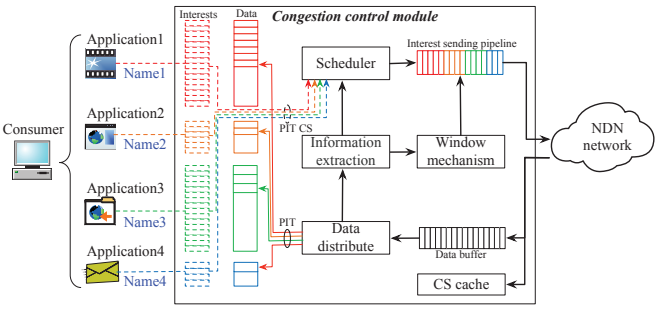


Fig. 3: DRL-CCP with multiple applications

contents to share the limited and dynamic bandwidth fairly (content-based), try to satisfy all the multiple requirements without overwhelming the network.

In order to achieve the above objective, we introduce a scheduler to work with the DRL-based window mechanism. As shown in Fig. 3, the goal of DRL-based model of window mechanism is to find the optimal decision policy of *cwnd* (which represents the overall throughput of consumer). The scheduler schedules multiple Interests according to various needs to fill the pipeline, the scheduling strategy can take into account user needs for different contents (such as priority), characteristics of requesting contents, network performance, fairness and so on. Both of the window mechanism and the scheduler are affected by the utility function defined later.

*2) **State Space**:* DRL is able to handle continuous, high-dimensional state space, but a high-dimensional state space would not only slow down convergence, and it also increases the computing load. So it makes sense to manually select the appropriate state variables that can represent the network states and capture the performance of actions. Here, we define the state space as a ten-dimensional variable as detailed in Table. I. Furthermore, to avoid the interference between sequential monitor intervals and satisfy the Markov property, we separate the monitor interval from the decision making interval and abandon the mixed part of any two monitor intervals.

TABLE I: state variable

variable	Definition
i_prefix	the prefix of requesting content
i_priority	the priority parameter of requesting content (introduced later)
i_cwnd	the immediate window of sending Interest packets
i_count	total number of Interest packets sent during the monitor interval
d_count	total number of Data packets received during the monitor interval
l_count	total number of packets retransmitted during the monitor interval
d_size	average size of Data packets received during the monitor interval
d_rtt	average RTT of Data packets received during the monitor interval
m_time	the time of monitor interval
d_time	the time of decision interval

*3) **Action Space**:* By referring to the actions of existing congestion control algorithms such as ICP, BBR and so on, we mainly control the sending rate of Interest packets by adjusting the size of *cwnd*, in response to variations in the network environments. And considering the convergence speed and the complexity of computing, here we can choose seven actions to simplify the action space of DRL-based model. As shown in Table. II, these actions are directly taken on the window size of sending Interest packets, the window remains unchanged until another action decision is calculated out and the current monitor interval finished. Finally, we find that these values can achieve satisfactory performance across different NDN network scenarios.

TABLE II: action options

Way of Change	Extent of Change (*cwnd*)
Increase	+1 , *1.25 , *1.5
Decrease	-1 , *0.75 , *0.5
Maintain	0

*4) **Reward**:* The definition of reward is important to DRL-CCP. Most existing works in congestion control of NDN treat all contents the same, that's unreasonable. To address various content characteristics and multiple user requirements, we design the single content's utility function as follow:

$$\begin{aligned} Utility_i(t) =& \alpha_i \cdot \log(throughput_i(t)) - \beta_i \cdot RTT_i(t) - \\ & \gamma_i \cdot loss_i(t) - \delta_i \cdot reordering_i(t) \end{aligned} \tag{1}$$

where $i \in \{1, 2, ..., N\}$ and $t > 0$ represent different types of contents and different monitor intervals. α_i, β_i, γ_i and δ_i are the parameters which control the relative weight or importance of throughput, average RTT, loss rate and packet reordering (it's represented by the occupancy rate of the sort buffer, which is used to store contents that can't arrive in the order of serial number and haven't been written to file) during current interval, they are relevant to user needs and content characteristics. That is, the control objective of a single content's request is to maximize the throughput while minimizing delay, loss rate and packet reordering. The single content's utility function is part of the total utility function below and also guides the scheduling strategy. The log function of throughput ensures the convergence of multiple contents when they share a bottleneck link. Then we define the total utility function of a consumer with different applications as follow:

$$Utility(t) = \frac{1}{N} \sum_{i=1}^{N} \omega_i \cdot Utility_i(t) \tag{2}$$

where ω_i is the priority parameter provided by us to distinguish different contents with different requirements, it can also be used as proportional allocation factor for the scheduler. The total utility function specifies the ultimate objective of congestion control in NDN, which is to maximize the long term total utility value of consumer, that high priority applications are consistently served first but low priority applications never starve, make full use of bandwidth resources to meet user requirements without causing congestion. In DRL, the action is evaluated based on a numerical reward, so we can make the reward equal to the total utility value in each monitor interval, and use the training algorithm to find the optimal strategy to maximize the long term reward.

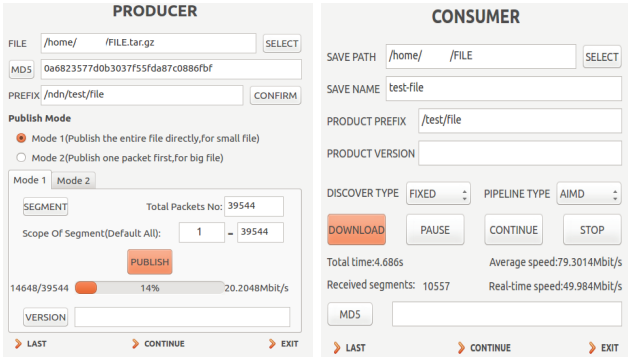
*5) **Training Algorithm**:* Generally, we can choose the specific model-free DRL training algorithm and design the structure of neural networks according to the computational capabilities of host devices. If computational capabilities is poor, the typical Deep Q-Network (DQN) model should be adopted, conversely we can choose the Deep Deterministic Policy Gradient (DDPG), Asynchronous Advantage Actor-Critic (A3C), etc. The former is suitable for discrete low-dimensional actions, while the latter can output action values in a continuous space. In particular, we can select discrete actions refer to the training results of the continuous output model. Also considering the convergence speed and the complexity of computing, we mainly introduce the classic DQN-based model to illustrate the efficiency of our proposed mechanism. Here we use S and A represent the state set and action set. The Q-value of each state-action combination (s, a)

is $Q(s,a)$, which reflects the quality of performing action a when the environment is in state s. The Q-values are updated every time an action a is taken in a state s, resulting in a reward $r{=}R_a(s,s')$ and a new state $s' \in S$ (Equation 3). So in this scene, we can make the reward $r{=}Utility(t)$ in each monitor interval. The Q-value will update toward the direction of maximizing the long term total utility value, and the optimal window control strategy will be found.

$$Q(s,a) = (1-\alpha)\,Q(s,a) \;+\; \alpha\,[r + \gamma\, max_{a'} Q(s',a')] \quad (3)$$

IV. Performance Evaluation

To demonstrate the performance of DRL-CCP, a series of experiments are conducted on the real-world test platform designed by ourselves. The test platform consists of two parts, the producer (Fig. 4a) used to upload contents and the consumer (Fig. 4b) used to request contents. We evaluate DRL-CCP's performance under some typical NDN transport scenarios, and compare it with two congestion control algorithms: ICP [13] and BBR-like algorithm [15]. These algorithms are implemented in the consumer side of the test platform.

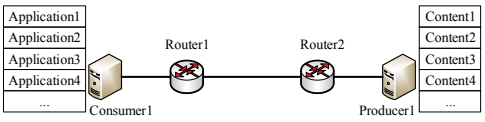


(a) Producer side (b) Consumer side

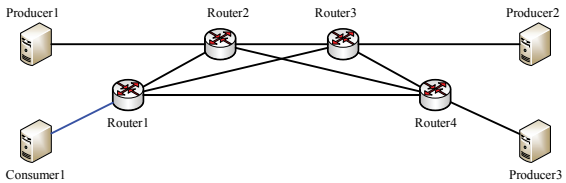
Fig. 4: Test platform

A. Experiment Settings

1) ***Topology****:* Our experiments are performed under the basic topology. Fig. 5 shows the network topology used in our experiments. Different transmission contents will be chosen, and different link bandwidths will be set according to the requirements of different scenarios.



(a) Baseline topology



(b) Multi-source and multi-path topology

Fig. 5: Experiment topology

TABLE III: Parameters setting

Parameter	Value
Underlying framework	tensorflow-1.7.0
Input dimension	10
Output dimension	7
Hidden layer	5 (128 nodes per layer, fully connected)
Activation function	relu
Optimizer	Adam
Learning rate	0.001
Discount factor	0.9

2) ***Parameters****:* Here we mainly introduce the classic DQN-based model. The parameters used in our experiments are summarized in Table. III. Using the appropriate parameter configuration, we get the optimal control strategy after about 2 hours of training, and the calculation time of calling the neural networks is about 3 to 5 milliseconds.

B. Performance in Different Scenarios

We evaluate DRL-CCP's performance under 3 representative NDN network scenarios, and are mainly concerned about the bandwidth utilization, delay, loss rate, packet reordering, the ability to respond to network changes, fairness and so on. Here, the bandwidth utilization is measured by the Data requesting rate (the Interest sending rate multiplied by the ratio of Data packet size to Interest packet size) of the consumer and the set bandwidth. Delay is the time between the consumer issues an Interest (the first time for a certain segment, since the Interest may experience retransmission) and receives the corresponding Data packet.

1) ***Fixed Bottleneck Bandwidth Network****:* We first form this baseline scenario in the baseline topology shown in Fig. 5a, only use Application1 to request Content1 here. The bottleneck bandwidth is 70Mbps (4Kbits per Data packet) and the delay on each link is 3ms.

The results are shown in Fig. 6, include the Data requesting rate and the boxplot of delay of the three methods in about 40s. It's clear to see that DRL-CCP can take full use of the set bandwidth, and it is more stable than the ICP and BBR-like algorithm. Due to the slow start and additive increase multiplicative decrease (AIMD) principle, the packets injected into the network during the initial phase of ICP will exceed twice the bottleneck bandwidth, which will cause congestion and increase the delay. And ICP halves the sending rate only according to the timeout event which is common in NDN networks, that is unresponsive and overreacted, and will cause a lot of dropping packets. Like ICP, BBR-like algorithm also uses slow start mode during the initial bandwidth detection phase. If the maximum instantaneous bandwidth measured continuously in 3 RTT cycles has only a small increase, it can be considered that the first measurement of the bottleneck bandwidth is completed, namely the packets injected into the network at this stage exceed the network capacity by about three times, which will lead instability. Especially in NDN network, it's hard to estimate the varied delay, so BBR-like algorithm is unable to make full use of bandwidth resources, and constantly detection may cause the jitter of delay. However, by continuous learning from experience and proactive controlling,

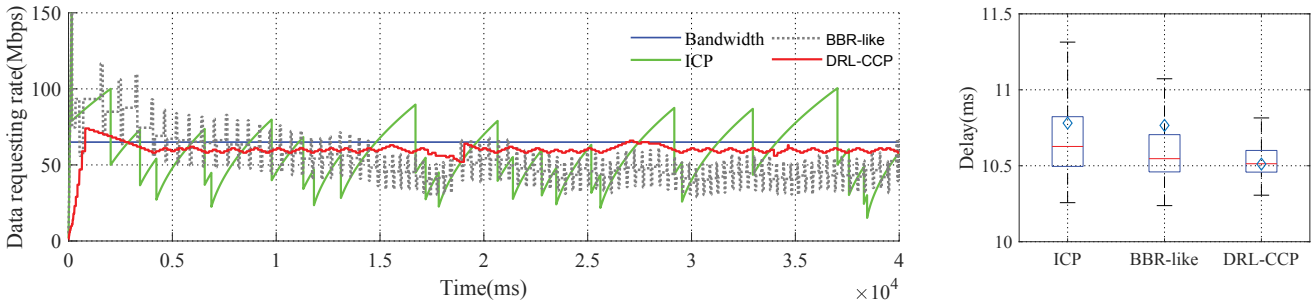


Fig. 6: Performance in fixed bottleneck bandwidth network(Data requesting rate and boxplot of delay)

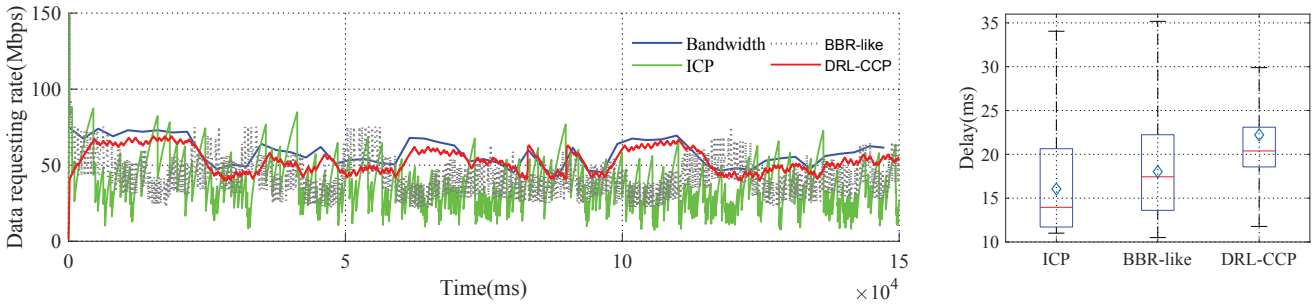


Fig. 7: Performance in approximate real NDN network(Data requesting rate and boxplot of delay)

DRL-CCP can reach the bottleneck bandwidth as quickly as possible in the initial stage, with small overshoot or even no overshoot in some cases. And it can maintain consistently high link utilization during the stable phase with low latency, packet loss rate and packet reordering.

*2) **Approximate Real NDN Network**:* In real NDN networks, the characteristics of multi-source and multi-path are ubiquitous, the link bandwidth, latency, and packet loss rate are dynamically changing. So we next extend our experiment to a more realistic scenario, where the bottleneck bandwidth varies significantly and the contents may arrive from multiple sources through multiple paths. Here the multi-source and multi-path topology (Fig. 5b) is used, the bandwidth and delay of all links are 80Mbps and 3ms, except the bottleneck bandwidth between Consumer 1 and Router 1 varies randomly between 40Mbps and 80Mbps. Producer1, Producer2 and Producer3 publish the same content for Consumer1.

The results are shown in Fig. 7. It's clear to see that even in this complex scenario, DRL-CCP still maintains high performance contrasts to ICP and BBR-like algorithm. In Fig. 7, DRL-CCP can dynamically follow the set value of bottleneck bandwidth to maintain high utilization of bandwidth, but the other two are either overreacting or unresponsive. And by comparing the delay in Fig. 7, it's easy to find that ICP and BBR-like have a wide range of delay fluctuations, but the delay of DRL-CCP are relatively stable within acceptable limits. Since the detection of RTT is more inaccurate in this scenario, the ICP and BBR-like algorithm become more unstable, either cause waste of resources or lead a lot of packet loss, and the packet reordering problem caused by the multi-source and multi-path characteristics of NDN is more severe. In short, our mechanism has the ability to quickly perceive and predict, cope with bandwidth variation (the same as varying packet loss rate or delay) and adapt to the characteristics of NDN.

*3) **Dynamic Behavior of Competing Contents**:* If congestion control algorithm is just designed to use link capacity aggressively, it will easily cause congestion and lead to unfairness. To show that DRL-CCP isn't aggressive and is able to realize content-based fairness which is on the basis of content characteristics and user requirements, we test four types of contents sharing the bottleneck link in this part. The topology shown in Fig. 5a is used, and four applications are running on Consumer1 to request the corresponding four types of contents, the bottleneck bandwidth between Router 1 and Router 2 changes as shown in Fig. 8, other parameters are the same as scenario 1. The priority parameters of the four applications are 3, 3, 4, 2. The scheduling strategy is based on proportional fairness, and then fine-tuned according to the single utility function. As shown in Fig. 8, DRL-CCP is able to

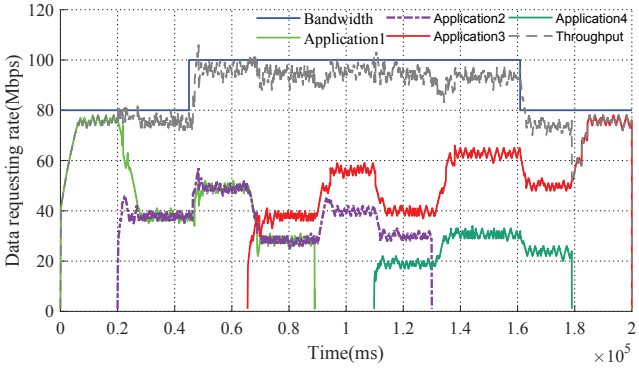


Fig. 8: Performance in competitive environment

rapidly converge to the fair sharing of link bandwidth according to similar priority parameters, even though the bottleneck link bandwidth is unstable. And the content-based fairness which is unique and applicable for NDN is realized, DRL-CCP can maintain high bandwidth utilization while sharing the bottleneck link bandwidth with different transmission speeds to meet different requirements.

In conclusion, DRL-CCP performs better than ICP and BBR-like mechanism both in stable and unstable environments. It can continuously make full use of bandwidth resources while maintaining lower delay, loss rate and packet reordering. Meanwhile, DRL-CCP can take advantage of the content-centric characteristic of NDN to achieve the desired objective of congestion control in NDN redesigned by us.

V. Conclusions

This paper proposes an intelligent congestion control mechanism based on DRL, which derives high-quality decision strategies by learning from the environment and experience, needs no preprogrammed rules or precise model of the dynamic networks. By appropriate design, optimization and training, DRL-CCP is applicable to diverse NDN network environments. It achieves optimal congestion control based on content fairness, especially fits the features of NDN and can better meet user needs. Lastly, the comparative experiments conducted on the test platform demonstrate the high performance of DRL-CCP.

References

[1] M. Dong, Q. Li, D. Zarchy, P. B. Godfrey, and M. Schapira, "PCC: Re-architecting congestion control for consistent high performance," in *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, (Oakland, CA), pp. 395–408, USENIX Association, 2015.

[2] C. Lim, Y. Kim, and I. Yeom, "Performance-based congestion control in information centric network," in *2017 International Conference on Information Networking (ICOIN)*, pp. 633–635, Jan 2017.

[3] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A survey of information-centric networking research," *IEEE Communications Surveys Tutorials*, vol. 16, pp. 1024–1049, Second 2014.

[4] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, CoNEXT '09, (New York, NY, USA), pp. 1–12, ACM, 2009.

[5] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, k. claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named data networking," *SIGCOMM Comput. Commun. Rev.*, vol. 44, pp. 66–73, July 2014.

[6] J. Li, S. Shi, Y. Ren, L. Li, and J. Zhi, "A content store-based module for congestion control algorithms of named data networking," in *2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pp. 253–259, Dec 2016.

[7] Y. Ren, J. Li, S. Shi, L. Li, and G. Wang, "An explicit congestion control algorithm for named data networking," in *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 294–299, April 2016.

[8] Z. Xu, J. Tang, J. Meng, W. Zhang, Y. Wang, C. H. Liu, and D. Yang, "Experience-driven networking: A deep reinforcement learning based approach," *arXiv preprint arXiv:1801.05757*, 2018.

[9] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[10] S. Mejri, H. Touati, and F. Kamoun, "Hop-by-hop interest rate notification and adjustment in named data networks," in *Wireless Communications and Networking Conference (WCNC), 2018 IEEE*, pp. 1–6, IEEE, 2018.

[11] J. Zhou, Q. Wu, Z. Li, M. A. Kaafar, and G. Xie, "A proactive transport mechanism with explicit congestion notification for ndn," in *Communications (ICC), 2015 IEEE International Conference on*, pp. 5242–5247, IEEE, 2015.

[12] W. Li, F. Zhou, K. R. Chowdhury, and W. M. Meleis, "Qtcp: Adaptive congestion control with reinforcement learning," *IEEE Transactions on Network Science and Engineering*, 2018.

[13] G. Carofiglio, M. Gallo, and L. Muscariello, "Icp: Design and evaluation of an interest control protocol for content-centric networking," in *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, pp. 304–309, IEEE, 2012.

[14] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "Bbr: Congestion-based congestion control," *Queue*, vol. 14, no. 5, p. 50, 2016.

[15] J. Huang, K. Lei, and Y. Wang, "Bbr based congestion control algorithm for ndn," *Journal of Chongqing University of Posts and Telecommunications(Natural Science Edition)*, vol. 1, p. 014, 2018 (in Chinese).

[16] S. Salsano, A. Detti, M. Cancellieri, M. Pomposini, and N. Blefari-Melazzi, "Transport-layer issues in information centric networks," in *Proceedings of the second edition of the ICN workshop on Information-centric networking*, pp. 19–24, ACM, 2012.

[17] L. Saino, C. Cocora, and G. Pavlou, "Cctcp: A scalable receiver-driven congestion control protocol for content centric networking," in *Communications (ICC), 2013 IEEE International Conference on*, pp. 3775–3780, IEEE, 2013.

[18] N. Rozhnova and S. Fdida, "An effective hop-by-hop interest shaping mechanism for ccn communications," in *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*, pp. 322–327, IEEE, 2012.

[19] 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 *ACM SIGCOMM Computer Communication Review*, vol. 43, pp. 55–60, ACM, 2013.

[20] K. Lei, C. Hou, L. Li, and K. Xu, "A rcp-based congestion control protocol in named data networking," in *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2015 International Conference on*, pp. 538–541, IEEE, 2015.

[21] K. Schneider, C. Yi, B. Zhang, and L. Zhang, "A practical congestion control scheme for named data networking," in *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, pp. 21–30, ACM, 2016.

[22] G. Carofiglio, M. Gallo, and L. Muscariello, "Joint hop-by-hop and receiver-driven interest control protocol for content-centric networks," *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 4, pp. 491–496, 2012.

[23] F. Zhang, Y. Zhang, A. Reznik, H. Liu, C. Qian, and C. Xu, "A transport protocol for content-centric networking with explicit congestion control," in *Computer Communication and Networks (ICCCN), 2014 23rd International Conference on*, pp. 1–8, IEEE, 2014.

[24] Y. Liu, X. Piao, C. Hou, and K. Lei, "A cubic-based explicit congestion control mechanism in named data networking," in *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2016 International Conference on*, pp. 360–363, IEEE, 2016.

[25] M. Dong, T. Meng, D. Zarchy, E. Arslan, Y. Gilad, B. Godfrey, and M. Schapira, "Pcc vivace: Online-learning congestion control," in *15th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 18). USENIX Association*, 2018.

[26] A. Narayan, F. Cangialosi, D. Raghavan, P. Goyal, S. Narayana, R. Mittal, M. Alizadeh, and H. Balakrishnan, "Restructuring endpoint congestion control," in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pp. 30–43, ACM, 2018.

[27] P. Bazmi and M. Keshtgary, "A neural network based congestion control algorithm for content-centric networks," *Journal of Advanced Computer Science & Technology*, vol. 3, no. 2, p. 214, 2014.

[28] T. Liu, M. Zhang, J. Zhu, R. Zheng, R. Liu, and Q. Wu, "Accp: adaptive congestion control protocol in named data networking based on deep learning," *Neural Computing and Applications*, pp. 1–9, 2018.