Peer-to-Peer Networking and Applications
https://doi.org/10.1007/s12083-019-00845-w

AFSndn: A novel adaptive forwarding strategy in named data networking based on Q-learning

Mingchuan Zhang [1] · Xin Wang [2] · Tingting Liu [1,3] · Junlong Zhu [1] · Qingtao Wu [1]

Received: 19 July 2019 / Accepted: 31 October 2019

Abstract
Named Data Networking (NDN) is a new network architecture, which employs a new content-centric communication model to replace the traditional host-centric communication model. In TCP/IP network, data packets are forwarded by routers according to routing table established previously. While in NDN, routing nodes can dynamically make forwarding decisions based on network status. By considering this forwarding feature, we proposed a novel adaptive forwarding strategy in Named Data Networking (AFSndn) based on Q-learning to minimize the delivery time. AFSndn is divided into two phases—Exploration phase and Exploitation phase. The Exploration phase aims to collect information, while the Exploitation phase aims to dynamically forward interest packets. Simulation experiment results show that AFSndn has better performance compared to others famous algorithms.

Keywords Named data networking (NDN) · Forwarding strategy · Q-learning

1 Introduction

Named Data Networking (NDN) [1–3] has been proposed as a new metric in the field of future Internet architecture. Since the architecture of NDN is different from that of traditional networks, the existing mechanisms or methods for traditional networks cannot be applied to NDN directly. Therefore, new methods should be studied for NDN, such as caching, forwarding, routing and congestion control [4–7], where network congestion is a major problem that needs to be solved. When NDN congestion occurs, its throughput will be reduced and the utilization of available resources will decrease. If forwarding packets can automatically keep away from congestion links, network congestions will be alleviated or even avoided.

In NDN, the forwarding packets not only refer to Forwarding Information Base (FIB) which is similar to IP routing table, but also consider the current network environment and the available communication interfaces of nodes [8]. We call this forwarding mechanism "smart forwarding", which enables routing nodes to make the forwarding decision in real time and dynamically. Meanwhile, the routing can adjust network load by controlling the size of their PIT [9–14]. And the routing will maintain a Pending Interest Table (PIT) to hold all not yet satisfied interest packets that were sent upstream towards potential data sources [15–17]. Through the states of interest packets to be processed and the interaction between interest packets and data packets, routing nodes can detect network congestions and dynamically select appropriate forwarding paths. Therefore, designing an efficient

This article is part of the Topical Collection: *Special Issue on Future Networking Applications Plethora for Smart Cities*
Guest Editors: Mohamed Elhoseny, Xiaohui Yuan, and Saru Kumari

✉ Xin Wang
wangxin@shisu.edu.cn

Mingchuan Zhang
zhang_mch@haust.edu.cn

Tingting Liu
liuttingyx@163.com

Junlong Zhu
jlzhu@haust.edu.cn

Qingtao Wu
wqt8921@haust.edu.cn

[1] Information Engineering College, Henan University of Science and Technology, Luoyang 471023, China

[2] School of Business and Management, Laboratory of Applied Brain and Cognitive Sciences, Postdoctoral Research Station, Shanghai International Studies University, Shanghai 200083, China

[3] The 32nd Research Institute of China Electronics Technology Group Corporation, Shanghai 201808, China

Published online: 07 January 2020

forwarding mechanism to actively avoid congestion paths can effectively alleviate the network congestion problem.

In this paper, we employ a transmission mode of one-interest-one-data to deal with above problem, where the data packets are returned in reverse direction along the forwarding path of interest packets. Therefore, if the delay time from sending an interest packet to receiving its data packet is shortest, the corresponding forwarding path selected by interest packet is perfect. If a node can choose a path with minimum delay time when forwarding interest packets, so as to alleviate or avoid network congestion. In consider of the forwarding characteristics and delivery time of data packets, we design an interest packet adaptive forwarding strategy in NDN (AFSndn) based on Q-learning algorithm. The Reinforcement Learning (RL) method was proposed by Watkins in 1995 [18]. The basic idea of the method is that agents interact with the environment autonomously. And then continuously accumulate the behavior data of each step to obtain the optimal action strategy. However, for standard RL, the agent is blind to search. This algorithm will make the target reward spread very slowly. Considering the drawback of slow learning in RL, AFSndn adds a heuristic knowledge function to the standard Q-learning algorithm. The algorithm uses prior knowledge as heuristic information to give additional rewards for specific actions, so as to accelerate the learning speed of agent in complex environment. The main contributions are as follows:

(1) We proposed an adaptive forwarding strategy based on Q-learning. This strategy introduces heuristic knowledge into the standard Q-learning algorithm, so that the agent can get rid of blind search to a certain extent to improve efficiency. In the process of forwarding, each routing node is regarded as an agent. It can automatically select forwarding paths and avoid congestion links.
(2) We consider the process of forwarding interest packets as two-stage decisions, i.e. Exploration stage and Exploitation stage, where each routing node needs to make a reasonable forwarding decision to achieve the best effect.
(3) The AFSndn strategy is evaluated by comparing with the MFC and RFA to verify its performance.

The rest of this paper is organized as follows. Section 2 gives a brief description of related work. Section 3 introduces the proposed AFSndn. Section 4 evaluates our AFSndn by simulations based on ndnSIM. Finally, the conclusion is presented in section 5.

2 Related work

In recent years, NDN has attracted wide attention of researchers. There is also a preliminary study on congestion control which is one of key technologies in NDN architecture. How to deal with the network congestion has become a focus in the field. This paper mainly studies the forwarding problem in DND.

The forwarding strategy can dynamically select one or more ports from the FIB for forwarding for each interest packet, which can make nodes use the connectivity between nodes efficiently. Carofiglio et al. [19] proposed a forwarding strategy based on the number of pending interest, where each port in the FIB entries records the corresponding value of pending interest, and calculates its weight based on the value of pending interest. Udugama et al. [20] proposed an effective and reliable real-time multipath forwarding strategy, where routing nodes simultaneously employ multiple ports to transmit interest packets according to the delay of ports. Lei et al. [21] proposed forwarding strategy based on probability entropy by combining dynamic port information and static routing information, where they employ entropy as a standard for port ordering and selecting. Rossini et al. [22] design a forwarding strategy considering the corresponding caching strategy because of the relationship of forwarding and caching. Li et al. [23] proposed a new flow control algorithm to ensure fairness for data packets and prevent network congestion. Qian et al. [24] proposed a probabilistic adaptive forwarding strategy based on ant colony algorithm, where they employ a statistical model to calculate the timeout parameter of the retransmission mechanism. Yi et al. [25] design an adaptive forwarding to retrieve data via the best performing paths, detect any packet delivery problems quickly and recover from them.

This paper introduces Q-learning into the forwarding process of the network. Each routing node is regarded as an agent, which improves the forwarding efficiency according to the information collected by itself. The standard Q-learning method can get the optimization result after enough learning rounds. However, the learning time for complex problems often exceeds the limits of tolerance. Therefore, the heuristic knowledge is injected into RL to improve the blind search method of RL. Andrew et al. [26] constructs the shaping function into the RL. And then the heuristic value is added to the agent's reward, which effectively improved the convergence speed. Bianchi et al. [27] proposed a heuristic function based on the mechanism of case-based reasoning. Before choosing the action, they first compare the similarity between the current status and the existing cases in the case base. And then choose the action under the guidance of the strategy of similar cases. Reinaldo et al. [28] design a class of algorithms called transfer learning heuristically accelerated RL that employs case-based reasoning as heuristics within a transfer learning setting to accelerate RL. Ferreira et al. [29] makes use of the concepts of modularization and acceleration by a heuristic function applied in standard RL to simplify and speed up the learning process of an agent that learns in a multi-agent multi-objective environment.

In this paper, we design an adaptive forwarding strategy, which based on the perception characteristics of the routing nodes. The whole forwarding process of the AFSndn is divided into two phases, namely Exploration phase and Exploitation phase. First, we collect information in the Exploration phase through a Q-learning with inspirational knowledge. Then in the Exploitation phase, the interest packages are dynamically forwarded based on the information gathered during the Exploration phase. Thereby alleviating the congestion problem of the network.

3 Proposed method: AFSndn

3.1 Overview for AFSndn

We consider each routing node in NDN as an intelligent agent. It not only has the ability to process information independently, but also can sense the state of the network.. With the change of time, routing nodes constantly perceive network states, and improve their intelligence by learning. We consider the routing process in NDN as a Markov decision process (MDP). In MDP, each agent has a finite state set S and an finite action set A. We introduced an evaluation function — Q function whose Q value is the maximum discounted cumulative reward when the process is beginning from state s with the first action a, where Q value is $Q(s, a)$, $s \in S$, $a \in A$. The learning of Q function values is done by the iteration of the Q value, where each iteration process updates a new Q value. Therefore, intelligent agents need to constantly interact with NDN to update all of the Q values. When all the values do not change much for an iteration process, the Q value function is considered to be convergent and the Q-learning is over.

Since the speed of standard Q-learning is rather slow, we employ a Q-learning algorithm with heuristic knowledge function $H : S \times A \rightarrow R$ to improve the learning speed of the agent. Under the guidance of prior knowledge, blind search can be turned into purposeful search. Assuming that there is no prior knowledge in the early stages of agent learning, the agent carries out perform a standard Q-learning process. In the process of learning, a high-level knowledge table is gradually established, so that the behavior choice of Q-learning has a certain bias under the guidance of knowledge. Here, the heuristic knowledge obtained in the initial stage is defined as the heuristic function $H(s, a)$. The heuristic function $H(s, a)$ is used to record the experience information about the interaction between the action a and the environment model is recorded under the state s. When the high-level knowledge table is roughly established, the Q-learning behavior of the lower level is guided. After enough rounds of learning, heuristic knowledge is gradually introduced to induce learning. The whole interaction process is shown in Fig. 1. And the loop process is shown in algorithm 1.

The Q function is stored and represented by a Lookup table whose size is the number of the cartesian element of $S \times A$. The heuristic function $H(s, a)$ is defined in the same way as the Q function, where $H(s, a)$ represents the H value of action A in state S. We apply the heuristic function $H(s, a)$ to action selection shown as

$$\pi(s_t) = \underset{a_t}{\operatorname{argmax}} \left[Q(s_t, a_t) + \delta H_t(s_t, a_t) \right]. \tag{1}$$

When performing selected actions, we observe the next state and receive the immediate enhancement signal. The updated formula of Q value is shown as

$$Q_{t+1}(s, a) \leftarrow (1 - a_n) Q_t(s, a) + a_n \left[r_t + \gamma \max_{a'} Q_t\left(s', a'\right) - Q_t(s, a) \right], \tag{2}$$

$$a_n = \frac{1}{1 + k_n(s, a)}, \tag{3}$$

where, s and a are the states and actions updated in the n-th cycle, $k_n(s, a)$ is the total number that state s and action a are visited in this n cycles (including the n-th cycle).

$$H\left(s', a\right) = \sum H(s, a) \tag{4}$$

The goal of AFSndn is to reasonably select the next hop node to minimize delay. Therefore, we think this process as an optimization problem at each node i as

$$\underset{a_i}{\text{minimize}}\ D_i(a_i) = D_i^v(a_i) + Z_v^d, \tag{5}$$

where, $D_i(a_i)$ indicates the delay from node i to destination for interest packets, a_i is the forwarding strategy adopted by node i, $D_i^v(a_i)$ is the delay from node i to its neighbor node v, Z_v^d represents the shortest delay from node v to destination d.

We assume that an interest packet is forwarded to the provider d passing through intermediate nodes i and v, where v is the next hop of node i. According to (2), we can get the update formula of Q value as

$$Q_i^{t+1} \leftarrow (1 - \omega(t)) Q_i^t + \omega(t) \left(D_i^v + \min D_v^d \right), \tag{6}$$

where, $\omega(t)$ is learning rate, D_i^v is the delay from node i to node v, $\min D_v^d$ is the shortest delay from node v to destination d.

From (6), we know that the calculation of the Q value of node i needs to obtain the corresponding information from the surrounding environment, that is D_i^v and $\min D_v^d$. We consider that the process of forwarding interest packets is composed of several interrelated decision stages. Every routing node needs to make reasonable forwarding decisions, so that the whole process can achieve the optimal results. In addition, we refer to the evolutionary algorithm [30] to divide the forward strategy into two phases, Exploration phase and Exploitation phase.

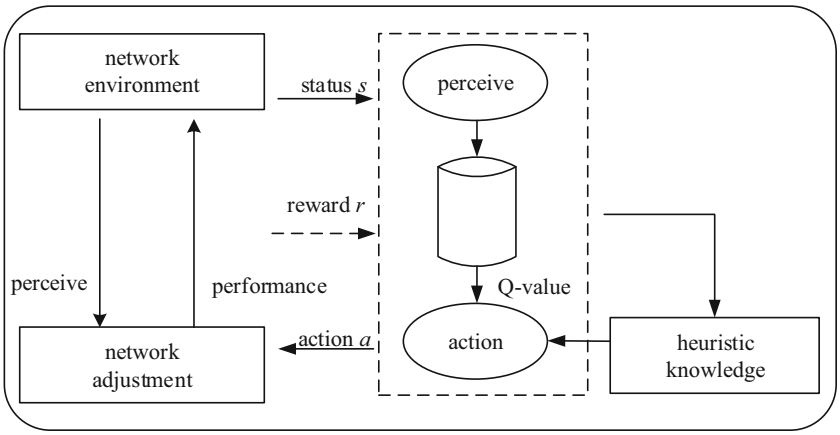


Fig. 1 Q-learning and network environment interaction process model

Algorithm 1: the Q-learning algorithm with heuristic knowledge

1 loop the routing node receives the request data;
2 Combine the heuristic function with the value function to select action a_t;
3 Perform action a_t, get reward $r(s,a)$, observe the next state s_{t+1};
4 Update the heuristic value $H(s,a)$;
5 Update the value function.
6 Update state $s_t \leftarrow s_{t+1}$
7 end loop;

3.2 Exploration phase—The information collection phase

Exploration phase aims to collect information. We calculate the Q value of the prefix-port pair <prefix, interface> based on the information carried by packets.

When a node needs to forward an interest packet, it first employs the longest prefix matching query FIB to obtain the list of candidate ports. And then the interest packet is forwarded through all candidate ports. We also need to modify the structure of FIB to support routing forwarding. Here, we add a new "Q-value" field to FIB, which is shown in Fig. 2. Meanwhile, we also need to modify the structure of the corresponding data packet as shown in Fig. 3. Firstly, we add a new field "minimum Q value" to record the minimum Q value corresponding to the prefix for packets. The minimum Q value is calculated by upstream nodes. Secondly, we add a new field "departure time" to record the time of the packet leaving the upstream node. This new field is used to calculate the transmission time from the upstream node to the downstream node. The calculation formula of Q value is shown in formula (7).

$$T = T_{now} - T_{departure}, \tag{7}$$

where, $T_{departure}$ is the departure time, T_{now} is the time to reach the downstream node.

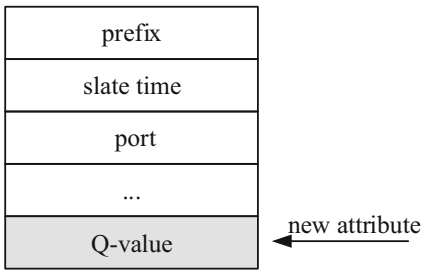


Fig. 2 Modified FIB table

Algorithm 1: Process of Exploration phase

1 while receive an interest packet;
2 if phase == Exploration phase;
3 for i=1; i<n do;
4 forwarding the interest packet to all candidate ports;
5 count++;
6 if count >= N_1;
7 phase = Exploitation phase;
8 count=0;
9 end if;
10 end for;
11 end while;

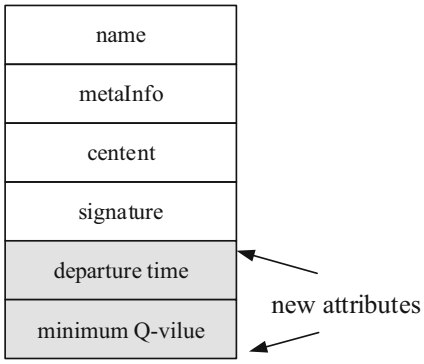


Fig. 3 Modified data packet structure

When a routing node receives data packets at a port, it calculates the Q value of the corresponding data stream at the port according to the information carried by data packets. And then it records the Q value to FIB. After N_1 interest packets are sent, the Exploration phase ends and then goes into the Exploitation phase.

3.3 Exploitation phase—The forwarding phase for interest packets

Exploration phase aims to forward interest packets. When a node forwards an interest packet, it will only select the optimal port to forward. We employ a probability approach to select the forwarding port, which may not be the port with the smallest Q value.

Algorithm 2: Process of Exploitation phase

```
while receive an interest packet;
  if Exploitation phase;
    select a port according to (8);
    sent an interest packet;
    count++;
      if | Q_min - Q_now | / Q_min > k;
        phase = Exploration phase;
        count=0;
      end if;
  end if;
end while;
```

The calculation method of forwarding probability is shown in formula (8).

$$P_j^f = \frac{k^{Q_j^f}}{\sum_v k^{Q_v^f}}, \tag{8}$$

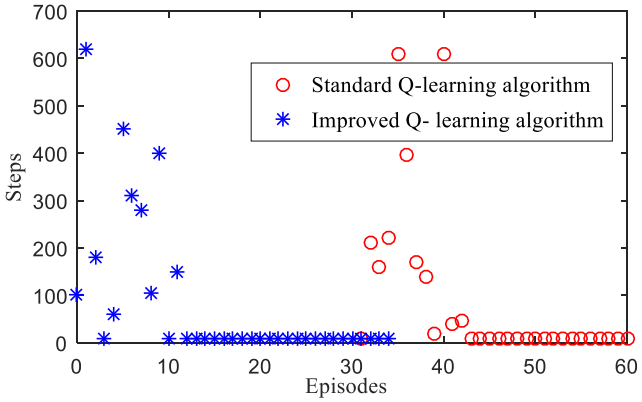


Fig. 4 Convergence of standard Q-learning algorithm and improved Q-learning algorithm

where, P_j^f is the probability to forward interest packets with prefix f through port j, Q_j^f is the Q value of data packets with prefix f through port j, k is a constant and $k > 0$. When the node receives a data packet, it calculates the Q value according to the information carried by the packet. The calculation of Q value is same as the Exploration phase.

When the condition of the formula (9) is satisfied, the Exploitation phase terminates and then the Exploitation phase restarts. Formula (9) is shown as

$$\frac{|Q_j^f - Q_j^{f'}|}{Q_j^f} > \theta, \tag{9}$$

where, Q_j^f is the minimum Q value, which is calculated in the Exploration phase, f is the prefix and j is the port. Q_j^f is obtained through constant calculation of updates in the Exploitation phase. The process of Exploitation phase is described as Algorithm3.

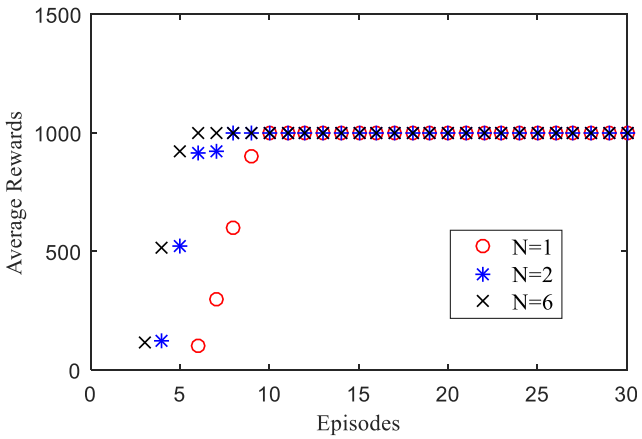


Fig. 5 Convergence of Q-learning algorithm with different number of agents

4 Simulation and analysis

We implemented the NDN protocol stack of NS-3 network simulator (http://www.nsnam.org/) by using the open source ndnsim [31] software packages, and evaluated the performance of the proposed AFSndn through simulation results. The ndnSIM simulation environment reproduces a basic structure of NDN nodes (i.e., CS, PIT, FIB, policy layer, etc.). The proposed enhancement learning method is implemented by MATLAB software. We employ C++ project which uses the MATLAB compiler to deploy algorithms. And then the C++ program is integrated with the ndnSIM environment to adjust in a simulated environment. In order to compare the difference between standard Q-learning and improved Q-learning. In each round of learning, we set the maximum number of search steps for agent to 700 steps. If the target point is not found after 700 steps, the search for this round is considered to have failed. Among them, Episodes refers to the learning period required by the agent from the initial state to the target state. Steps refers to the sum of the steps required by the agent from the initial state to the target state.

Figure 4 shows the convergence between standard Q-learning algorithm and improved Q-learning algorithm. The standard Q-learning algorithm achieves convergence at the 43rd cycle, while the Q-learning algorithm with heuristic knowledge achieves convergence at the 13th cycle. It can be seen from this that the Q-learning algorithm with heuristic knowledge can quickly converge to the optimal state and shorten the learning time of agents. Moreover, the Q-learning algorithm injected with heuristic knowledge expands the application space of RL and can be applied in more complex environments.

The goal of Q-learning is to trade off Exploration (untrained) and Exploitation (trained). We set a reward value to assess agents. In the process from an initial state to a target state, if an agent reaches to the target state, we give the agent a reward with value +100; if the agent encounters a congested link, we give the agent a reward with value −10. Agents are evaluated by the average of obtained reward values for a period of time. Figure 5 shows the convergence of Q-learning algorithm with different number of agents, $N=1$, $N=2$ and $N=6$. It can be seen from Fig. 5 that the number of agents has certain influence on the convergence speed of Q-learning algorithm. For example, the Q-learning algorithm reaches to convergence at the 12th cycle with $N=1$, while the algorithm reaches to convergence at the 9th cycle with $N=2$ and at the 5th cycle with $N=6$, respectively. The main reason is that the Q-learning algorithm makes agents have higher learning ability. With the increase of the number of agents, the more shared knowledge is obtained among agents, which makes agents reach to the target state faster.

Our experiments focus on evaluating the ability of AFSndn to stabilize network conditions and adapt multiple paths. Therefore, we employ a representative network topology shown in Fig. 6. And then compare average delay and packet loss rate of AFSndn with those of MFC [23] and RFA [19] .

Figure 7 shows average delays of AFSndn by increasing link failure rate in comparison with MFC and RFA. Form Fig. 7, average delays of three algorithms increase with deteriorating of link congestion. The average delay of MFC algorithm is 32% and 27% higher than that of AFSndn algorithm and RFA algorithm, respectively. The main reason is that

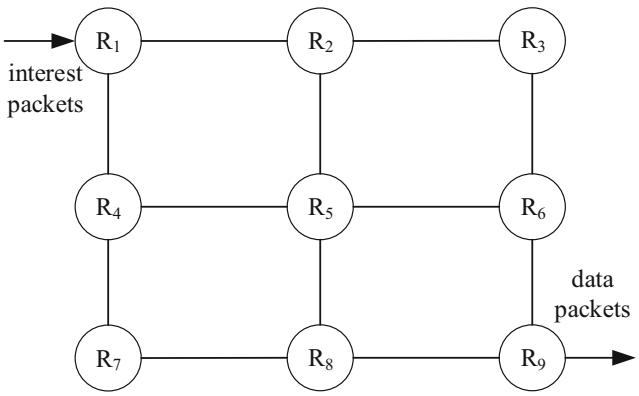


Fig. 6 Evaluation topology

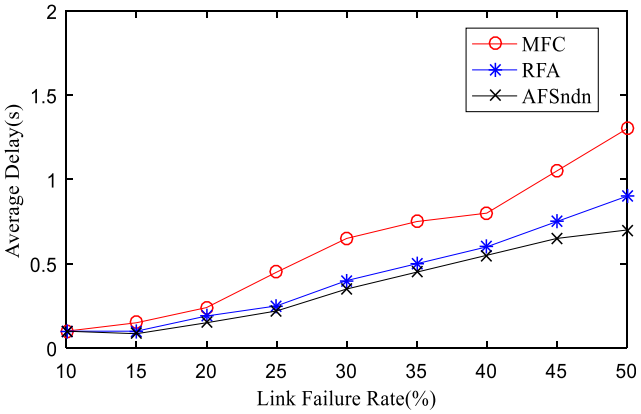


Fig. 7 Average delay by increasing link failure rate

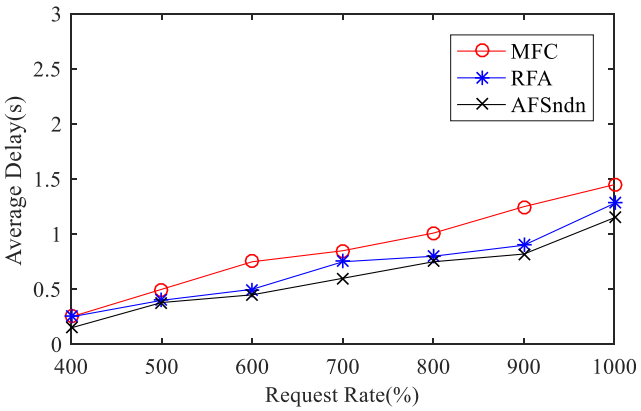


Fig. 8 Average delay by increasing request rate

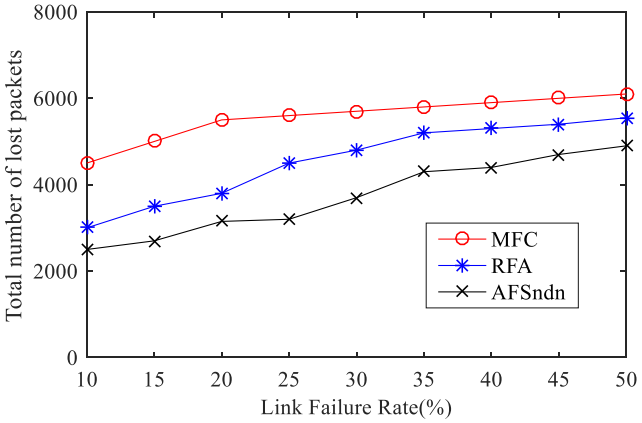


Fig. 9 Losses of packets by increasing congestion rate

MFC algorithm does not control the rate of the forwarding interest packages. It leads to the excessive number of returned packets. At this point, the routing node will discard the packets that cannot be processed, which increases the latency of the data request.

Figure 8 shows the average delays of AFSndn by increasing request rate in comparison with MFC and RFA, where the overall congestion level is set to 35%. Form Fig. 8, the average delay of MFC is the largest, while those of RFA and AFSndn are rather small.

Figures 9 and 10 show the losses of packets by increasing congestion rate and request rate, respectively. The losses of packets of MFC, RFA and AFSndn will increase with increasing both link congestion and users' requests. Furthermore, the loss of packets of MFC is the largest, while losses of packets of RFA and AFSndn are rather less. The main reason is that MFC will rigidly increase network load with increasing users' requests, and then aggravate network congestion. The RFA algorithm can select the best link, which is used to forward interest packets if network congestion occurs. If the optimum link encounters congestion, the data packets passing by this link will be discarded. AFSndn can dynamically avoid congested links based on network status collected to reduce the losses of packets.

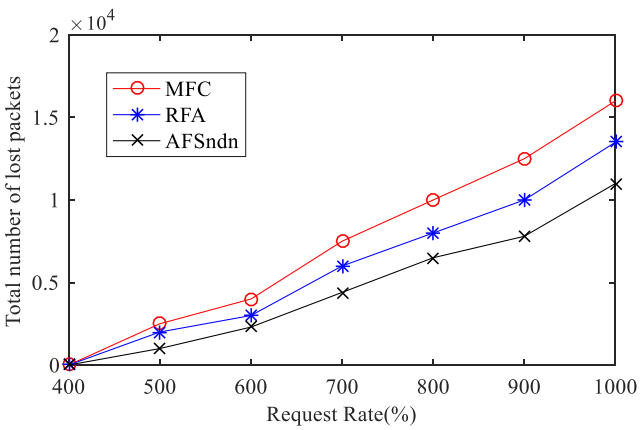


Fig. 10 Losses of packets by increasing request rate

5 Conclusion

According to the forwarding characteristics of interest packets in NDN, we proposed an adaptive forwarding strategy based on Q-learning with heuristic knowledge. This strategy regards each routing node as an agent, which establishes and maintains a forwarding table, and selects forwarding objects according to the table. Each routing node needs to make reasonable forwarding decisions, so that the whole process can achieve an optimal result. Finally, experiments show that the AFSndn can avoid the problematic path actively, reduce the round-trip delay of data requests and alleviate (or avoid) network congestion. The simulation results show that AFSndn has high transmission efficiency and better stability.

Acknowledgements This work is partially supported by the National Natural Science Foundation of China (NSFC) under Grants no. U1604155 and no. 61871430, and in part by the China Postdoctoral Science Foundation under Grant no. 2018 M630461, and in part by the Science Foundation of Ministry of Education of China under Grants No. 19YJC630174, and in part by the Scientific and Technological Innovation Team of Colleges and Universities in Henan Province under Grants No.20IRTSTHN018, and in part by the basic research projects in the University of Henan Province under Grants No. 19zx010.

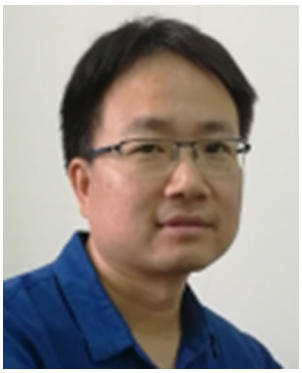
Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interests regarding the publication of this paper.

References

1. Fang C, Yu F, Huang T et al (2015) A survey of green information-centric networking: research issues and challenges. IEEE Commun Surv Tutorials 8(3):1455–1472
2. Amadeo M, Campolo C, Quevedo J et al (2016) Information-centric networking for the internet of things: challenges and opportunities. IEEE Netw 30(2):92–100
3. Zhang L, Afanasyev A, Burke J et al (2014) Named data networking. ACM SIGCOMM Comput Commun Rev 44(3):66–73
4. Feng B, Zhang H, Zhou H et al (2017) Locator/identifier Split networking: a promising future internet architecture. IEEE Commun Surv Tutorials 19(4):2927–2948
5. Araújo FRC, de Sousa AM, Sampaio LN (2019) SCaN-Mob: An opportunistic caching strategy to support producer mobility in named data wireless networking. Comput Netw 156:62–74
6. Kumar N, Aleem A, Singh AK, Srivastava S (2019) NBP: Namespace-based privacy to counter timing-based attack in named data networking. J Netw Comput Appl 144:155–170
7. Zhang H, Quan W, Hu B et al (2016) Smart identifier network: a collaborative architecture for the future internet. IEEE Netw 30(3): 46–51
8. Acs G, Conti M, Gasti P, Ghali C, Tsudik G, Wood CA (2019) Privacy-aware caching in information-centric networking. IEEE Trans Dependable Secure Comput 16(2):313–328
9. Liu T, Zhang M, Zhu J et al (2018) ACCP: adaptive congestion control protocol in named data networking based on deep learning. Neural Comput Appl 31:4675–4683

10. Pacifici V, Dán G (2016) Coordinated selfish distributed caching for peering content-centric networks. IEEE/ACM Trans Networking 24(6):3690–3701
11. Song F, Ai Z, Li J, Pau G, Collotta M, You I, Zhang H (2017) Smart collaborative caching for information-centric IoT in fog computing. Sensors 17(11):2512
12. Feng B, Zhou H, Zhang M et al (2015) Cache-filter: a cache permission policy for information-centric networking. KSII Trans Int Inf Syst 9(12):4912–4933
13. Zhang M, Xie P, Zhu J et al (2017) NCPP-based caching and NUR-based resource allocation for information-centric networking. J Ambient Intell Humaniz Comput (4–5):1–7
14. Li Q, Lee P, Zhang P et al (2017) Capability-based security enforcement in named data networking. IEEE/ACM Trans Networking 25(5):2719–2730
15. Karami A (2015) ACCPndn: adaptive congestion control protocol in named data networking. J Netw Comput Appl 56(1):1–18
16. Qiao X, Ren P, Chen J, Tan W, Blake MB, Xu W (2019) Session persistence for dynamic web applications in Named Data Networking. J Netw Comput Appl 125:220–235
17. Yang H, Wang X, Yang C, Cong X, Zhang Y (2018) Securing content-centric networks with content-based encryption. J Netw Comput Appl 128:21–32
18. Ben J, Kröse A (1995) Learning from delayed rewards. Robot Auton Syst 15(4):233–223
19. Carofiglio G, Gallo M, Muscariello L et al. (2013) Optimal multipath congestion control and request forwarding in information-centric networks, IEEE international conference on network protocols (ICNP)
20. Udugama A, Zhang X, Kuladinithi K et al. (2014) An On-demand Multi-Path Interest Forwarding Strategy for Content Retrievals in CCN, IEEE/IFIP Network Operations and Management Symposium (NOMS), pp. 1–6
21. Lei K, Wang J, Yuan J (2015) An entropy-based probabilistic forwarding strategy in named data networking. IEEE international conference on communications (ICC), pp. 5665–5671
22. Rossini G, Rossi D (2014) Coupling caching and forwarding: benefits, analysis, and implementation. ACM international conference on Information-centric networking, pp. 127–136
23. Li C, Huang T, Xie R et al. (2015) A novel multi-path traffic control mechanism in named data networking. IEEE international conference on telecommunications, pp. 60–66
24. Qian H, Ravindran R, Wang G et al. (2013) Probability-based adaptive forwarding strategy in named data networking. IFIP/IEEE international symposium on integrated network management, pp. 1094–1101
25. Yi C, Afanasyev A, Moiseenko I et al (2013) A case for stateful forwarding plane. Comput Commun 36(7):779–791
26. Ng AY, Harada D, Russell S (1999) Policy invariance under reward transformations: theory and application to reward shaping [C]. Sixteenth international conference on machine learning, pp. 278–187
27. Bianchi RAC, Celiberto LA, Santos PE et al (2015) Transferring knowledge as heuristics in reinforcement learning: a case-based approach. Artif Intell 226:102–121
28. Bianchi RAC, Santos PE, Silva IJ, Celiberto LA, de Mantaras RL (2018) Heuristically accelerated reinforcement learning by means of case-based reasoning and transfer learning. J Intell Robot Syst 91(2):301–312
29. Ferreira LA, Costa Ribeiro CH, Augusto DCBR (2014) Heuristically accelerated reinforcement learning modularization for multi-agent multi-objective problems. Appl Intell 41(2): 551–562
30. Yogeswaran M, Ponnambalam SG (2012) Reinforcement learning: exploration–exploitation dilemma in multi-agent foraging task. OPSEARCH 49(3):223–236
31. Mastorakis S, Afanasyev A, Moiseenko I, et al. (2015) ndnSIM2.0: A new version of the NDN simulator for NS-3. Technical report NDN-0028



Mingchuan Zhang He was born in Henan Province, PRC in May 1977. Mingchuan Zhang studied in Beijing University of Posts and Telecommunications (Beijing, PRC) from September 2011 to July 2014, majored in Communication and information system and earned a Doctor of Engineering Degree in three years' time. He works as an Associate Professor in Henan University of Science and Technology. His research interests include bio-inspired networks, Internet of Things, future Internet and computer security.

Xin Wang She was born in Heilongjiang Province, PRC in April 1980. Xin Wang studied in Beijing University of Posts and Telecommunications (Beijing, PRC) from September 2013 to July 2017, majored in management science & engineering and earned a Doctor of Engineering Degree in four years' time. She works in Shanghai International Studies University from Mar 2017 to now. Her research interests include communication management, and communication security.

Tingting Liu She was born in Henan Province, PRC in Oct 1992. Tingting Liu received her M.S. degree from Henan University of Science and Technology, China, in 2018. She works in The 32nd Research Institute of China Electronics Technology Group Corporation from July 2018 to now. Her research interests include information center network, cloud computing and Internet of Things.

Qingtao Wu He was born in Jiangsu Province, PRC in Mar 1975. Qingtao Wu studied in East China University of Science and Technology (Shanghai, PRC) from Mar 2003 to Mar 2006, majored in computer application and earned a Doctor of Engineering Degree in three years' time. He works as a Professor in Henan University of Science and Technology. His research interests include component technology, computer security and future Internet security.

Junlong Zhu He received the Ph.D. degree in computer science and technology from the Beijing University of Posts and Telecommunications (Beijing, PRC) in 2018. In 2018, he joined the Henan University of Science and Technology, Luoyang, China, where he is currently a Lecturer with the Information Engineering College. His current research interests are focused theoretical and algorithmic issues related to on large-scale optimization, distributed multi-agent optimization, stochastic optimization, and their applications to machine learning, signal processing, communications and networking.