A fair dynamic content store-based congestion control strategy for named data networking

Yan-Hong Liu, Feng-Hua Huang and Hua Yang

College of Information Science and Engineering, Shanxi Agricultural University, Taigu, People’s Republic of China

ABSTRACT
In this paper, the congestion control for named data networking (NDN) is studied. A novel dynamic content store-based congestion control strategy is proposed on account of the characteristic of in-network cache in NDN. A queuing network model is constructed to judge whether congestion occurs. If the network has the tendency of congestion or the congestion happened, the buffer of the output queue is dynamically expanded by borrowing NDN content store (CS), and the forwarding rates of data packets and corresponding interest packets are reduced so as to prevent or alleviate network congestion. In order to reflect fairness, the CS to be borrowed by the data output queue in the port is calculated in terms of the data output queue length and its weight. The simulation results based on ndnSIM show that the given scheme improves the bottleneck link utilization and maintains a low packet loss rate and average flow completion time.

1. Introduction
With the development of network interfaces such as ‘Ethernet’ or ‘Wi-Fi’, the architecture of current host-centric internet protocol (IP) network is facing inefficiencies such as IP address exhaustion, insufficient scalability, poor security and mobility (Ren et al., 2016). To solve these problems caused by IP address, name data networking (NDN) has been proposed by adopting a content-centric paradigm, where data is uniquely identified by a name for addressing and caching (Xia & Xu, 2013). The NDN architecture offers a number of attractive advantages such as network load reduction, low dissemination latency and energy efficiency. Meanwhile, this architecture is easy lead to congestion due to large amount of redundant information caused by multi-path forwarding strategy and the strict one-to-one relationship between interest packet and data packet (Lei & Yuan, 2014; T. Li & Wang, 2019; Zou, 2019). Therefore, how to solve network congestion has become one of the key problems in NDN architecture (Jiang & Luo, 2018; Shen et al., 2019; Yao et al., 2017).

With the expansion of network scale, all kinds of contradictions and problems of IP network for address-centric have highlighted. To solve these problems, the communication mode of information-centric network (ICN) has evolved from host–host to host–network and the forwarding mechanism has converted from traditional storage-forwarding to buffer-forwarding, which solves the efficient transmission of massive information. Among them, the NDN is the most representative network architecture. It focuses more on the content name, which can effectively solve a series of problems caused by the IP address. In recent years, the problem of NDN congestion control has received significant attentions. According to the role responsibility for reacting to network congestion, the NDN congestion control schemes can be broadly classified into two categories: implicit congestion control and explicit congestion control (C. C. Li et al., 2017). In implicit congestion control scheme, the requester judges the occurrence of network congestion through the retransmission timeout mechanism (RTO), and the network congestion is controlled or alleviated by reducing the sending rate of interest packets. In explicit congestion control scheme, congestion occurrence is detected by the intermediate node, then the congestion information is explicitly feedback to the receiver, the receiver adjusts the sending rate of interest packets to control the return rate of data packets (Shi et al., 2016). Implicit congestion control scheme is suitable for end-to-end communication transmission protocol. The NDN architecture advocates the pull-based paradigm where in-network caching and multi-path
forwarding are pervasive and the data source cannot be determined, which makes it very difficult to estimate timeout, so the implicit congestion control scheme is difficult to implement in NDN. In explicit congestion control scheme, the congestion state is detected by the intermediate node of the router and fed back to the downstream node or receiver to adjust the forwarding rate of interest packets (data packets). This kind of congestion control scheme is suitable for multi-path forwarding strategy and has attracted a lot of attention of scholars (Zhang et al., 2016). A congestion control algorithm is proposed by routing the content request to the potential content source (Huang et al., 2017). Considering the impact of link delay and interruption, a minimum-delay congestion control algorithm combined with reinforcement learning is given to realize the intelligent forwarding (Y. D. Wang et al., 2020). In reference W. J. Wang and Luo (2018), the weight of the data stream is dynamically adjusted according to its forwarding rate. The token bucket algorithm is adopted to slow down the forwarding rate of data stream exceeding the fair rate, and the congestion information is fed back to the downstream node through the explicit feedback mechanism. Based on the characteristic of in-network cache in NDN, a store-based congestion control algorithm is designed to solve the burst flows by considering the interaction between router buffer size and congestion control mechanism in Xia and Xu (2013).

As aforementioned, the research on NDN congestion control mostly focuses on solving or alleviating network congestion by reducing the forwarding rate of interest packets, and the utilization of in-network caching in NDN is rarely considered. On the other hand, when the characteristic of intermediate node deploying cache is considered, the fairness of cache resource allocation has not been adequately analyzed in designing NDN congestion control scheme. Therefore, combined with the existing interest control protocol (ICP), according to the characteristic of in-network cache in NDN, the problem of designing fair dynamic content store-based congestion control scheme for NDN has not been adequately addressed and remains an interesting and challenging research topic. This situation motivates the present control strategy study of ICP and dynamic content store (ICP + DCs).

In this paper, we are concerned with the fair dynamic store-based congestion control problem for NDN. By assuming that the router configures the same buffer size for each port, the NDN in-network cache is allocated fairly to the ports according to the data stream queue length and its corresponding weight. If the queue control threshold is less than the given constant, the in-network cache is borrowed by the port and the forwarding rates of data packets and corresponding interest packets are reduced to preprocess congestion. When in-network cache is used by the port, the queue control threshold is recalculated. If the updated threshold is less than the given value, the forwarding rates of interest packets corresponding to all data streams in the port are reduced. The main contributions of this paper can be highlighted as follows: (1) the NDN in-network cache and its dynamic allocation are considered to smooth the burst flow and alleviate network congestion; (2) the network congestion can be judged in advance so as to make preprocessing; (3) a fair dynamic content store-based congestion control algorithm for NDN is designed.

This paper is organized as follows. In Section 2, the fair dynamic content store-based congestion control algorithm is introduced. In Section 3, the performance of the algorithm is analyzed according to the simulation results. Finally, conclusion is drawn and the possible research directions in the future are pointed out in Section 4.

2. Algorithm design

2.1. Motivation

According to the characteristic of in-network cache in NDN, by borrowing the router cache to expand the port buffer, a more effective and practicable congestion control algorithm may be obtained. On the other hand, the state forwarding plane in NDN routers makes per-interest forwarding state available at each hop, which provides native support for flow-based control. Note that the terminology ‘flow’ represents the stream of interest packets or data packets. From the discussions above, it is clear that content store-based flow-aware control mechanisms are natural candidates for congestion control in NDN architecture when we seek to realize better data delivery performance.

If the network has the tendency of congestion or the congestion happened, the buffer of data output queue is dynamically expanded by borrowing NDN content store (CS), so as to smooth the burst flow and prevent the network congestion. In order to reflect fairness, the CS to be borrowed is calculated in terms of the data output queues length and its weights.

2.2. Dynamic interactive cache congestion control model

By adding modules of data table (DT) and content store-based rate adjusting (CRA), the dynamic interactive cache congestion control algorithm, which is executed in NDN, is proposed based on the original NDN routing model. The type of DT is first in first out (FIFO). If the data packet
cannot be forwarded immediately, the control information of this data packet is recorded in an entry of DT. When the data packet is forwarded, the corresponding entry in DT is deleted. The dynamic interactive cache congestion control model is shown in Figure 1.

2.3. The judging network congestion model

Assuming that each port of the router is configured with the same buffer $B$. Let $T^i(t)$ is the queue control threshold of port $i$, when NDN cache is not borrowed by port $i$, it is given by the following formula:

$$T^i(t) = \alpha \left( B - Q^i(t) \right)$$  \hspace{1cm} (1)

where $\alpha$ is a positive constant describing control parameter, $Q^i(t) = \sum_{j=1}^{N} l^j_i(t)$ is the queue length of output data in port $i$, $l^j_i(t)$ is the output queue length of data stream $j$ in port $i$, $N$ is the number of data streams in router.

Let $B^i$ be the borrowed buffer size from NDN by all ports in the router, when the NDN cache is borrowed by the port $i$, $T^i(t)$ is replaced by

$$T^i(t) = \alpha \left( B + B^i_j - Q^i(t) \right)$$  \hspace{1cm} (2)

where $B^i_j(t)$ represents the buffer size borrowed from NDN by port $i$ and satisfies $B^i_j = \sum_{j=1}^{M} B^i_j(t)$, its calculation formula is to be given later; $M$ is the number of port in router.

When the port $i$ does not borrow buffer from NDN, the value of $T^i(t)$ is obtained by (1). In this case, if $T^i(t)$ is less than or equal to the given constant $\beta_B$, this indicates that the network is congested. Then, the forwarding rates of interest packets corresponding to all data packets in the port $i$ are reduced to alleviate the congestion. If the value of $T^i(t)$ is greater than $\beta_B + \alpha B^i_j(t)$, it means that the network congestion has been alleviated, and the port $i$ no longer borrows NDN cache.

2.3.1. Determine the weight of data stream

Let $\omega_j(t)$ is the weight of data stream $j$, it is given as follows:

$$\omega_j(t) = \frac{p(l^j_i(t))}{s(\theta_j)}$$  \hspace{1cm} (3)

where $p(l^j_i(t))$ and $s(\theta_j)$ are monotonically decreasing nonnegative functions, $l^j_i$ is the output queue length of data stream $j$, $\theta_j$ is the priority value of data stream $j$. For the convenience of calculation, let $p(l^j_i(t)) = 1/l^j_i(t)$ and $s(\theta_j) = 1$, then

$$\omega_j(t) = \begin{cases} 
\frac{1}{l^j_i(t)} & l^j_i(t) > 0 \\
0 & l^j_i(t) = 0 
\end{cases} \hspace{1cm} (4)$$

The above formula indicates that if the output queue length of data stream $j$ is equal to zero, its corresponding weight is zero and the data stream $j$ will not be forwarded. If the output queue length of data stream $j$ is greater than zero, the corresponding weight decreases with the increase of the output queue length of the data stream $j$. If all data streams have same priorities, the increase of the output queue length of the data stream $j$ can reduce the weight of the data stream so as to reduce the output rate of the data stream. This can prevent greedy users from seizing network resources and smooth burst flow.

2.3.2. Calculate the fair buffer allocated by NDN to the port

Let $B^i_j(t)$ is the buffer that the NDN allocates to the port $i$ at time $t$, it is given as follows:

$$B^i_j(t) = \frac{\sum_{j=1}^{N} l^j_i(t) \omega_j(t)}{\sum_{j=1}^{N} \sum_{j=1}^{M} l^j_i(t) \omega_j(t)}$$  \hspace{1cm} (5)

where $l^j_i = \sum_j l^j_i$. The buffer size borrowed from NDN by port $i$, which is related to the output queue length of the data stream and its corresponding weight, is dynamic. It should be pointed out that, to embody fairness, the weight value is given in terms of the output queue length of all data streams in the router.
2.3.3. Fair output queue length
Based on the assumption that each port in the router is configured with the same buffer capacity \( B \), neglecting the priority of data stream, the fair output queue length of the port \( i \) can be given by \( q_i(t) = B / \sum_{j=1}^{N} f(j) \). If the length of the data stream \( j \) in the port \( i \) is equal to zero, let \( f(j) = 0 \); If the length of the data stream \( j \) in the port \( i \) is greater than zero, let \( f(j) = 1 \).

2.4. Algorithm implementation
The pseudo code of the fair dynamic content store-based congestion control algorithm is given as follows:

**Algorithm 1** Pseudo-code of the fair dynamic store-based congestion control algorithm
BEGIN
input: Data;
PIT match;
if(there are matching entries in PIT)
{
CS insert;
for i from 1 to M
{
if(\( Q_i(t) \leq B \))
{
if (\( T_i(t) \leq \beta_B \))
{enable fair cache, reduce the sending rate of data packets and corresponding interest packets;}
}
else
{
if(\( T_i(t) \leq \beta_c \))
{reduce the send rate of all interest packets;}
forwarding data packets;
}
}
if(there is no matching entry in PIT)
{packet loss;}
}
END

The detailed execution process of CRA, which is the key part to solve congestion, is given in the above algorithm. By using the algorithm, for possible congestion trends, the sending rate of data packets and corresponding interest packets are reduced to prevent network congestion. In the case of network congestion, the sending rate of all data packets is reduced to alleviate network congestion. In the following, the effectiveness of the proposed algorithm is given.

### Table 1. Setting of experimental parameters.

| Name                  | Value                  |
|-----------------------|------------------------|
| Working flow          | \( C_1 - S_1 \)        |
| Burst flow            | \( C_2 - S_2, C_3 - S_2, C_4 - S_2, C_5 - S_2 \) |
| Interest packet size  | 40 B                   |
| Data packet size      | 1040 B                 |
| Link bandwidth        | 100 Mbps               |
| Bottleneck link bandwidth | 10–60 Mbps           |
| Link delay            | 10 ms                  |

3. Simulation results and analysis
In this section, we study the performance of the proposed algorithm using the ndnSIM simulator (Alexander et al., 2012). To this end, we carried out extensive simulations and comparisons with the methods of ICP and ICP+DCs under a variety of network scenarios. In what follows, we present a selected set of simulation results to illustrate the benefits of our scheme.

For what concerns the topology used for the evaluations, we consider the single bottleneck link topology shown in Figure 2, which is typically used for congestion control analysis. There are two servers and five consumers. The consumers send 3000 interest packets per second to servers. The link propagation delay is 10 ms. The bottleneck bandwidth ranges are from 10 Mb/s to 60 Mb/s. The experimental parameters are shown in Table 1.

In the experiment, ICP, which is a classical interest request rate control algorithm in NDN, is adopted. On the basis of ICP algorithm, ICP+DCs are adopted in case of burst flow. With the increasing of bottleneck link width from 10 to 60 Mb/s, the simulation results of bottleneck link utilization and packet loss rate are shown in Figures 3 and 4.

The comparison results of bottleneck link utilization rate are described in Figure 3. It is easy to see that the bottleneck link utilization rate of ICP decreases from 86% to 78% with the increase of bottleneck bandwidth. The reason is that ICP uses congestion window to forward interest packets. As a result, the queue length increases with the rise of interest packets so as to reduce of bottleneck link utilization. By using the ICP+DCs, the bottleneck link utilization rate increases gradually and approaches 94% with the widening of bottleneck bandwidth. The reason is that ICP+DCs dynamically adjusts the cache allocation according to the output queue length of the whole data streams in the router and allocates fair bandwidth for each data stream. This algorithm sufficient utilizes more resources. As a result, it relaxes the data stream and alleviates the congestion so as to better use the network bottleneck link bandwidth to some extent. Compared with ICP algorithm, ICP+DCs adds the functions of predicting congestion and enabling NDN fair cache, which makes it
better adapt to the change of bottleneck link bandwidth and improve bandwidth utilization.

As can be seen from Figure 4, compared with ICP, ICP+DCs is lower packet loss rate, which decreases rapidly with the increase of bottleneck link bandwidth. The reason is that when there is a trend of congestion, ICP+DCs algorithm enables NDN fair cache to reduce the forwarding rate of data stream and corresponding interest stream for output queues exceeding the fair length, so as to reduce data packet loss and alleviate congestion. The ICP judges the network congestion by the timeout of data packet transmission, which delays the processing of congestion and results in high packet loss rate.

The change trends of average flow completion time of ICP and ICP+DCs are showed in Figure 5. With the increase of the number of flows, the average flow completion time of the two algorithms become longer, but the average stream completion time of ICP+DCs is shorter. When the number of flows increases to 50, the average flow completion time increases significantly. The reason is that the network is congested at this time. The slope of ICP+DCs curve gradually tends to be smooth after 75
streams, which indicates that ICP+DCs can better adapt to network changes.

4. Conclusion and prospects

In this paper, we have investigated the congestion control problem for NDN. In terms of the character of in-network cache of NDN, the buffer of data output queue is dynamically expanded by borrowing NDN CS. The buffer size to be borrowed by port is calculated on the basis of the data output queue length and its weight. Combined with the existing congestion control algorithm ICP, a novel fair dynamic content store-based congestion control algorithm is designed. Simulations are conducted to validate the effectiveness and efficiency of our proposed scheme, and the results demonstrate the benefits in improving network performance, such as link utilization, packet loss rate and average completion time of data stream.

In the proposed algorithm, quantitative evaluating the fairness and optimizing the control thresholds are not considered, which will be the future research direction. On the other hand, more influencing factors, such as the popularity of network resources, the number of resource visits, can be considered to detect the network congestion, which may be another research direction in the future.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

Alexander, A., Moiseenkko, I., & Zhang, L. X. (2012). ndnSIM, NDN simulator for NS-3. Named data networking (NDN) project. University of California.
Huang, S., Hao, Y. M., Jiang, L. H., & Zheng, D. L. (2017). Congestion control mechanism based on adaptive forwarding over naming data networking. Computer Applications and Software, 34(12), 184–188. http://doi.org/10.3969/j.issn.1000-386x.2017.12.035
Jiang, Y. M., & Luo, J. T. (2018). Redundant control strategy in named data networking. Application Research of Computers, 35(10), 3048–3052. http://doi.org/10.3969/j.issn.1001-3695.2018.10.039
Lei, K., & Yuan, J. (2014). Content distribution mechanisms of named data networking. Telecommunications Science, 30(9), 27–36. http://doi.org/10.3969/j.issn.1000-0801.2014.09.005
Li, C. C., Xie, R. H., Huang, T., & Liu, Y. J. (2017). Jointly optimized congestion control, forwarding strategy and link scheduling in a named-data multihop wireless network. Frontiers of Information Technology & Electronic Engineering, 18(10), 1573–1590. https://doi.org/10.1631/FITEE.1600158
Li, T., & Wang, F. H. (2019). Advantages and feasibility analysis of tactical networks based on NDN. Information & Communications, (6), 66–68. http://doi.org/10.3969/j.issn.1673-1131.2019.06.031.
Ren, Y. M., Li, J., Shi, S. S., Li, L. L., Wang, G. D., & Zhang, B. C. (2016). Congestion control in named data networking-A survey. Computer Communications, 86(3), 1–11. https://doi.org/10.1016/j.comcom.2016.04.017
Shen, Y. X., Wang, X. W., Li, J., & Zeng, R. F. (2019). Multi-source based end-node congestion control algorithm for NDN. Cyberspace Security, 10(8), 75–83. http://doi.org/10.3969/j.issn.1674-9456.2019.08.012
Shi, S. S., Ren, Y. M., & Li, L. L. (2016). A content store-based congestion control algorithm for named data networking. Chinese High Technology Letters, 26(4), 359–366. http://doi.org/10.3772/j.issn.1002-0470.2016.04.005
Wang, W. J., & Luo, J. T. (2018). Active congestion control strategy in named data networking. Computer Engineering and Applications, 54(10), 115–120. http://doi.org/10.3778/j.issn.1002-8331.1612-0302
Wang, Y. D., Zhang, Y., Chen, Y. X., & Zhang, Y. (2020). Research on active congestion control mechanism in named date network. Manned Spaceflight, 26(1), 69–75. http://doi.org/10.16329/j.cnki.zrht.2020.01.010
Xia, C. M., & Xu, M. W. (2013). Survey of information-centric networking. Journal of Frontiers of Computer Science and Technology, 7(6), 481–493. http://doi.org/10.3778/j.issn.1673-9418.1303025
Yao, J. F., Yin, B. Q., Tan, X. B., & Bao, Y. Z. (2017). A game theoretic framework for congestion control in named data networking. Information Technology and Control, 46(4), 605–618. https://doi.org/10.5755/j01.ict.46.4.15680
Zhang, C. C., Wang, L., Lv, W., & Xia, L. (2016). Content centric network congestion control strategy for multi-service. Computer Engineering, 42(4), 79–82. http://doi.org/10.3969/j.issn.1000-3428.2016.04.015
Zou, G. J. (2019). Research on congestion control mechanism of name data networking. Beijing University of Posts and Telecommunications.