Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2021) 29: 1124 – 1138
© TÜBİTAK
doi:10.3906/elk-2005-146

PFECC: a precise feedback-based explicit congestion control algorithm in named data networking

Hui Li*
School of Computer and Communication, Lanzhou University of Technology, Lanzhou, P.R. China

Received: 21.05.2020 • **Accepted/Published Online:** 09.10.2020 • **Final Version:** 30.03.2021

Abstract: Named data networking (NDN), as a future Internet architecture that is a promising alternative to TCP/IP networks, has the new features of connectionless, in-network cache, and hop-by-hop forwarding, which makes the congestion control algorithms of traditional TCP/IP networks unable to be directly applied to NDN. In addition, since the optimal size of the sending window cannot be determined, the existing window-based congestion control algorithms generally use the AIMD-like window adjustment algorithm, which cannot achieve the optimal throughput. In this paper, we propose a precise feedback-based, multipath-aware congestion control algorithm PFECC, which is inspired by Accel-Brake Control algorithm. PFECC considers the influence of Interest flows, uses a fair queuing algorithm at the intermediate node to calculate the target rate of each flow, and gives accurate feedback on each dequeued data packet. The consumer changes the size of the sending window according to the feedback to quickly converge to the target bandwidth. To fully exploit the hop-by-hop adaptive forwarding feature of NDN, each downstream node timely senses the congestion trend of the upstream link to split the forwarding rate of Interest to avoid congestion. Simulation results show that PFECC can effectively reduce transmission delay, treat each flow fairly, and converge to the best throughput faster.

Key words: Named data networking, congestion control, multipath forwarding, precise feedback

1. Introduction

Information services have become the main body of today's Internet services, and it is getting impossible for the TCP/IP architecture, which is based on end-to-end communication to realize efficient, secured, and large-scale sharing and distributing of content. Being employed to cope with those challenges, named data networking (NDN) [1] has attracted much attention as a future Internet network that is the most promising alternative to the current TCP/IP architecture. When a new network architecture appears, how to effectively use limited network resources to improve the quality of service becomes one of the key issues that need to be solved.

NDN changes the conversational paradigm of the traditional TCP/IP protocol, focusing on uniquely named data instead of IP addresses. NDN adopts the consumer-driven "pull" data transmission mode. The consumer sends interest packet(s) containing the requested content name, and the source (producer) with the corresponding content returns the data packet along the "bread crumb". In order to facilitate data sharing and make effective use of network resources, the returned data packets are also cached in the content store (CS) of the router on the path as long as possible. If another consumer requests the content, the data packets cached in the node along the path can replace the producer to respond. This method effectively improves the efficiency

*Correspondence: lihuiceo@163.com

of data retrieval, but it also leads to uncontrollability of the content source, which brings many challenges to the congestion control mechanism of NDN.

The traditional TCP/IP network detects congestion by measuring round-trip time (RTT), but in NDN, due to its connectionless, multisource, multipath, Interest packets aggregation, and other factors, the RTT taken from the consumers varies greatly, so the estimated retransmission timeout(RTO) as an indicator of NDN congestion makes the detection of congestion unreliable. The uncertainty of RTT also makes it impossible to determine the optimal sending window size of the window-based congestion control algorithm, as the sending window needs to be determined based on bandwidth and RTT. The result is that window size control algorithms such as additive increase multiplicative decrease (AIMD) must be used, resulting in window-based congestion control algorithms not achieving the optimal throughput.

Furthermore, active queue management (AQM) solutions such as RED [2] and CoDel [3] are often utilized as sections of the congestion control algorithms, they can send congestion signals (through explicit congestion notification or packet loss) and perform congestion control before the bottleneck link buffer is filled, thereby reducing delay and avoiding congestion. However, the AQM scheme does not inform the consumer(s) about how to increase the sending rate. When the available bandwidth increases, consumer(s) have to blindly increase the rate again, which will lead to congestion and insufficient link utilization.

In this paper, inspired by the idea of Accel-Brake Control (ABC) [4], we propose an explicit multipath-aware congestion control algorithm based on precise feedback for NDN referred to as PFECC. The intermediate node calculates each flow's target bandwidth and adds feedback to each data packet according to the dequeue rate. The consumer adjusts the size of the sending window accurately according to the feedback. At the same time, intermediate nodes monitor the congestion trend of upstream links to implement multipath forwarding to avoid congestion.

To verify the effectiveness of PFECC, we have conducted relevant experiments on ndnSIM [5], a simulator based on ns-3, and compared it with other congestion control algorithms. The results show that PFECC can effectively reduce packet transmission delay and increase network throughput.

The contributions of PFECC are summarized as follows:

(1) A new target bandwidth allocation mechanism, inspired by ABC but different from it.

(2) A new feedback mechanism feeds back the most severe congestion to consumers without increasing the link load.

(3) A new multipath forwarding mechanism avoids congestion effectively and achieves higher throughput.

This paper is organized as follows: Section 2 makes a brief summary of NDN's congestion control algorithm. Section 3 proposes the congestion control algorithm PFECC. We put the evaluation in Section 4 and finally summarize this paper in Section 5.

2. Related work

The currently proposed NDN congestion control algorithms can be divided into two types: window-based control algorithms and rate-based control algorithms. In the window-based congestion control method, the window size defines the maximum number of Interest packets that can be sent. Inspired by TCP/IP congestion control mechanism, ICP [6] sets an RTO timer for each Interest packet and uses a TCP-like AIMD window adjustment algorithm, which does not consider the multisource problem in NDN. To solve the multisource problem, CCTCP [7] maintains an RTO value for each source. It uses the "anticipated interests" mechanism to estimate the retransmission timeout to trigger the correct RTO value. Remote adaptive active queue management (RAAQM) [8] uses the router label to determine the transmission path of each flow, and uses the change in RTT as the congestion detection method. HR-ICP [9] adds hop-by-hop Interest shaping mechanism based on ICP. These implicit congestion control methods based on RTO timeout cannot accurately reflect the degree of network congestion, and have a certain lag in congestion control. This prompted the emergence of explicit congestion notification methods. Explicit Control Protocol [10] divides the degree of congestion into three levels, and the intermediate node explicitly feeds back the congestion level information to the receiver. At the same time, the receiver uses the multiplicative increase additive increase multiplicative decrease (MIAIMD) algorithm to adjust the sending window of Interest packets. In the CHoPCoP [11] scheme, a queue length threshold is set for the router. Only when the queue length is greater than this threshold, will it start hop-by-hop control through the fair share interest shaping (FISP) mechanism. It uses the AIMD window adjustment mechanism at the receiver. PCON [12] detects link congestion by the sojourn time of data packet in the queue, and explicitly feeds back to the receiver. The receiver controls the sending rate of its interest packets by using the AIMD-like algorithm, but there is an unnecessary rate drop at the intermediate node. In the rate-based congestion control algorithm, Hop-by-hop interest shaping mechanism (HoBHIS) [13, 14] calculates the forwarding rate of interest based on the queue occupancy and the available resources of each flow. The flow control in HoBHIS is entirely inside the network, and the receiver only performs actions in response to a clear backpressure signal from the network. Hop-by-hop interest shaping (HIS) [15] considers that the size of the interest packets is also a factor that affects link congestion, and according to the constraint that the sum of the rate of interest and the data should be less than the bandwidth, the optimal rate of interest adjustment is calculated, but HIS does not implement flow fairness. The rate-based [16] scheme directly informs the receiver of the rate of interest, takes the allowable sending rate of each interface as a weight, and selects the interface through the roulette algorithm to achieve multipath forwarding.

In addition, congestion control based on multipath forwarding has also been the subject of a lot of research work. Based on RAAQM [8], Carofiglio et al. further added multipath and adaptive forwarding to intermediate nodes to optimize congestion control. In [17], the authors proposed a multipath forwarding strategy CF based on pending interest (PI), which calculates its weight based on the number of PIs per interface and uses a weighted round-robin strategy to select the interface. Another PI-based scheme is the PI strategy, which selects the interface with the smallest PI number to forward the interest packets. Multipath-aware ICN rate-based congestion control [18] is a rate-based multipath-aware congestion control. It uses path ID to identify the transmission path of the flow, and uses specific hints of subflows containing in interest packets to guide the data packets along the path of the subflow.

3. Design of PFECC congestion control algorithm

PFECC is a window-based congestion control protocol. When congestion happens suddenly, the window-based control mechanism reacts faster than the rate-based control mechanism [19]. As shown in Figure 1, PFECC can be divided into three functional modules: active queue management (AQM) module, consumer window adjustment module and multipath forwarding module. The AQM module is responsible for detecting link congestion and providing backpressure signals; the consumer window adjustment module adjusts the sending rate of interest packets based on the feedback; the multipath forwarding module is responsible for dynamically splitting the interest traffic when the link is already congested or is about to congest to reduce or avoid congestion.

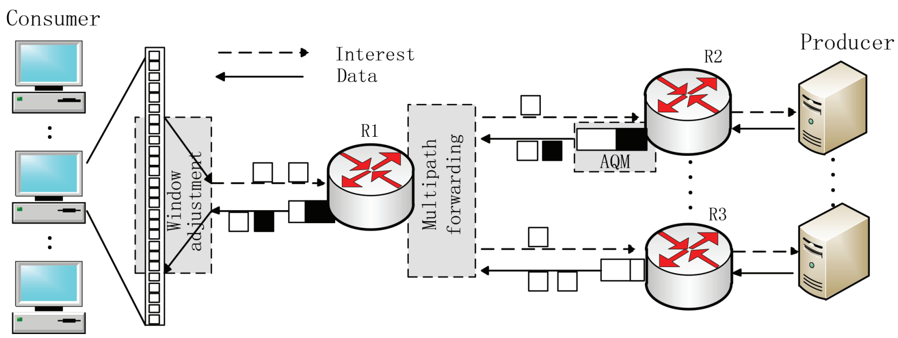

Figure 1. The model of PFECC.

3.1. Congestion detection

In NDN, congestion manifests as interest packets or data packets overflowing in the interface buffer, but the current buffer queue length cannot be regarded as congestion, because burst traffic may cause the queue to become longer. The method of congestion detection can be achieved by detecting the size of the router's Pending interest table (PIT) table or the average length of the interface queue. Nichols and Jacobson pointed out that the most reliable place to detect congestion is where it occurs [3], so we use a relatively simple congestion detection method in the egress queue of the PFECC router, that is, if the data packet stays in the queue for more than 5 ms (CoDel's goal is also to keep the minimum delay of packets in the queue below 5 ms), it can be regarded as congestion. PFECC uses different feedback method, the specific details are explained in the next section.

3.2. Explicit feedback

(1) Calculate the target rate of each flow Before calculating the target rate, we must first define the concept of NDN flow. In the TCP/IP network, the flow is defined as a 5-tuple: (source address, destination address, source port, destination port, transmission protocol), and there is no concept of end-to-end transmission in NDN, so the flow cannot be defined as a 5-tuple. In this paper, we define a flow as a set of interests and their corresponding returned data, which are forwarded to the router under the same forwarding information base (FIB) entry at a given time.

NDN routers forward interests and data through the interface, so each interface should have interest flows and data flows at the same time (as shown in Figure 2), and each flow establishes a queue in the buffer of the interface. These queues are logical queues, meaning that all queues may be located in a physical buffer. In NDN, the name of the interest packets may be very long because it contains transaction information of many applications. During the transmission process, it will consume a part of the link bandwidth that cannot be ignored. Therefore, both interests and data contribute to congestion. And in any interest shaping algorithm, their interdependence must be considered.

Considering the above factors, this paper considers the influence of interest flows when calculating the target bandwidth of each flow. The router calculates the target rate of each flow as follows:

$$tr_i(t) = \frac{\alpha U(t) - A_i(t)^{-1} U(t)(x_i(t) - d_t)^{+}}{N_i(t) + M_i(t) \times s} \tag{1}$$

In formula (1), $U(t)$ is the link bandwidth, $x_i(t)$ is the observed queuing delay, and d_t is the preset

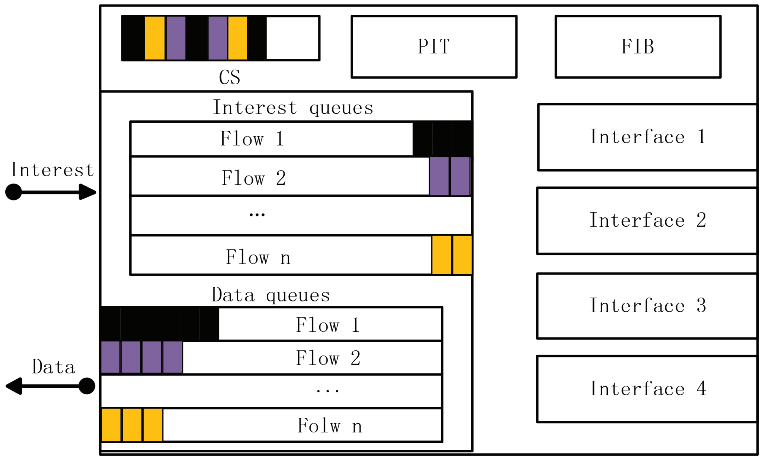

Figure 2. Flow queues in NDN router.

delay threshold. Here we set d_t as the value of sojourn time which is considered the congestion time of the data packet in the queue. α is a constant close to 1, used to control the stability and performance of the algorithm. $A_i(t)$ is the average response time for the router to forward the interest packet and the corresponding data packet to the router. In this paper, the exponential weighted moving average is used to calculate the average response time, $A_i(t) = \theta \times A_i(t)_{sample} + (1-\theta) \times A_i(t)$, where θ is the weighting factor. Y^+ is a function, which is equal to $max(y, 0)$. $N_i(t)$ and $M_i(t)$ are the number of interest flows and data flows at the current time, and s is the average size ratio of interest packet and data packet.

The above formula has the following explanation: When the queue delay is small, $x_i(t) - d_t < 0$, then $(x_i(t) - d_t)^+ = 0$, and all flows share the link bandwidth fairly. Note that α is a constant close to 1 (for example 0.95). The reason why the total target rate is designed smaller than the bandwidth is to sacrifice a small portion of the bandwidth in exchange for a significant reduction in queue delay. When $x_i(t) - d_t > 0$, the second item of the formula is to make the queue empty in time $A_i(t)$. In the ABC algorithm, a fixed time is used to empty the queue, and it is proved that the algorithm is stable when the fixed time is greater than $2/3$ times the round trip time delay. Because the data can be obtained from multiple sources or caches in NDN, the round trip time is uncertain, so in this scheme, the queue is emptied within the time $A_i(t)$.

(2) Feedback mark After calculating the target rate of each flow, to make each flow reach the target rate $tr_i(t)$, the router should calculate the packet share $f_i(t)$ that should be marked as "add" for each flow. Since PFECC is a protocol that uses data packet as a selective acknowledgment of interest packet, the current dequeue rate of data packets on the router can provide an accurate prediction of the incoming rate of future data packets (i.e. an RTT). Thus, the PFECC router uses the dequeue rate of the packet to compare with the target rate to calculate the packet share $f_i(t)$ marked as "add". Suppose that the current dequeue rate of flow i in the router is $cr_i(t)$. If the packets with the $f_i(t)$ share are marked as "add", this will bring an average of $2f_i(t)$ packets. Because in the PFECC scheme, each data packet marked as "add" returns to the consumer, the consumer's sending window will be increased by 1 (see 3.3). After an RTT, the enqueue rate of packets to this router will be $2f_i(t)\,cr_i(t)$. To reach the target rate $tr_i(t)$, the calculation of $f_i(t)$ should be $2f_i(t)\,cr_i(t) = tr_i(t)$, then,

$$f_i(t) = \left\{ \frac{1}{2} tr_i(t)\, cr_i(t)^{-1}, 1 \right\} \tag{2}$$

PFECC uses the measured dequeue rate $cr_i(t)$ and target rate $tr_i(t)$ to recalculate $f_i(t)$ on each dequeued data packet using formula (2), and make explicit feedback on the data packet, so that consumers can react to link utilization on time. PFECC uses the probability $f_i(t)$ to mark outgoing packets. This simple method has lower time complexity and space complexity and can ensure that the packets marked as "add" do not exceed the share of $f_i(t)$. In addition, compared with the ACK method, the backpressure mechanism based on the piggybacking of the data packet can accurately and timely feed back the congestion information to the consumer without increasing the network overhead, further improving the efficiency of the congestion control algorithm.

3.3. Consumer window adjustment algorithm

The consumer adjusts the size of the interest packet sending window in a timely and accurate manner according to the feedback mark carried in the data packet. In the PFECC scheme, the consumer maintains a sending window for each flow. The specific approach is that once the packet marked as "add" is received, the consumer increases its sending window value by 1. Therefore, the data packet marked as "add" will cause 2 interest packets to be sent. Once the packet marked as "reduce" is received, the consumer decreases its sending window value by 1. Therefore, the packet marked as "reduce" will cause no new interest packet to be sent. In this way, although the change in the value of the sending window is small each time, it can cause the size of the sending window to change between 0 and 2 times the current value within an RTT. In addition, to ensure the fairness between the flows, the PFECC consumer's sending window for each flow is also increased by 1 per RTT [4].

PFECC makes a feedback on each data packet and slightly changes the size of the sending window accurately at the consumer, avoiding the sending window to swing back and forth between the maximum and minimum values, reducing the transmission delay and achieving better throughput than AIMD window adjustment algorithm.

3.4. Multipath forwarding

The above congestion control method will achieve satisfying result on a single path, but in NDN, the routing mechanism is separated from the forwarding plane, and the router can adaptively forward interest traffic on multiple interfaces. Therefore, if the upstream link is congested, the router can adaptively select the forwarding interface according to the state of the link interface, which is beneficial to make full use of network resources and avoid further congestion. This section implements a multipath forwarding strategy based on the upstream interface congestion trend. The strategy reuses the existing packet queuing delay information to predict the interface congestion trend by calculating the delay trend. The congestion trend of the interface is used to split the interests' traffic of the downstream nodes to realize the multipath forwarding strategy.

This algorithm assumes that the available interface information for a given prefix of interest has been given by the routing protocol in the FIB. Here the split ratio of the interests is determined only based on the status of each interface. Interests are forwarded according to the rank of the interfaces in the FIB. That is, if the number one interface can meet the forwarding requirements of interest traffic, only this first interface is used. If the first interface is going to be congested or has congested, part of the traffic on the first interface will be transferred to the remaining interface(s) until all the interfaces are congested. The multipath forwarding strategy is described as follows:

(1) Determine the queuing delay trend This strategy calculates the queuing delay trend of the received data packets within a fixed time interval (see Table 1 for symbols and descriptions). Suppose that the queuing

delays of the received m data packets are $D_1, D_2, D_3 \cdots D_m$, respectively. Here we set a minimum queuing delay threshold $MF\_minDelay$ for multipath forwarding. When one of the received queuing delay values is lower than $MF\_minDelay$, all interest traffic is still forwarded along the original path. The reason for setting $MF\_minDelay$ here is that if there is an upward trend in queuing delay, the multipath forwarding strategy will be used even if there is no congestion. This causes the forwarding path switching to be too frequent, which is not conducive to the stability of the multipath forwarding algorithm, and frequent path switching will reduce the network throughput. However, if the value of $MF\_minDelay$ is greater than the congestion sojourn time (5 ms), the response of the multipath algorithm will be lagged, so $0 < MF\_minDelay < 5$ ms, the specific value of $MF\_minDelay$ will be determined in the experiment. If the received queue delay values are all greater than $MF\_minDelay$, then the delay trend judgment function can be used:

$$S' = \frac{\sum_{i=2}^{m} Q\left(D_i \geq D_{i-1}\right)}{m-1} \qquad \left(D_i > MF\_minDelay\right) \tag{3}$$

From the function definition of S', we know that $0 \leq S' \leq 1$. If the value of D_i is random, the value of S' swings around 0.5. If the value of D_i has a strong downward trend, the value of S' is close to 0. If the value of D_i has a strong upward trend, the value of S' is close to 1.

Table 1. Symbols and descriptions.

Symbols	Descriptions
D_i	Delay for data packet in the queue
M	Number of received data packets
$MF\_minDelay$	Minimum delay threshold for multipath forwarding
S'	Delay trend
$P_{reduction}$	Reduced forwarding ratio of current interface
P_{change}	Adjust the interface forwarding scale factor
$P\left(f\right)$	Forwarding ratio of current interface
$P\left(f^{'}\right)$	Forwarding ratio of the remaining interfaces
N	Total number of currently forwarded interfaces

(2) Adjust the interface forwarding ratio After calculating the queuing delay trend of the upstream interface queue, the downstream node will quantify the forwarding ratio of interests for each interface. For each FIB prefix, PFECC maintains a forwarding ratio for each interface. For the first-ranked interface, the initial forwarding ratio is 100%, and for other interfaces, the initial forwarding ratio is 0%. After the downstream node obtains the congestion trend S' of the upstream interface, quantify the forwarding ratio of each interface as:

$$P_{reduction} = P\left(f\right) \times \left(S^{'} - 0.5\right) \times P_{change} \tag{4}$$

$$P\left(f\right) = P\left(f\right) - P_{reduction} \tag{5}$$

$$P\left(f^{'}\right) = P\left(f^{'}\right) + \frac{P_{reduction}}{N-1} \tag{6}$$

P_{change} is a fixed value, which is used to adjust the forwarding ratio of the interface. Through experiments, we found that a value of 1%–3% is more suitable. The above formula has the following explanation: If the queue of the upstream interface is already congested or is about to congest, then the queuing delay of the data packets in the queue will have an upward trend, then $S^{'} > 0.5$. According to formula (4), the value of $P_{reduction}$ is positive, indicating that part of the current interest traffic needs to be transferred to other interfaces, to reduce the load of the current link and congestion. Once the upstream queue congestion is reduced or disappeared, the queue delay of the data packets in the queue will have a downward trend, then $S^{'} < 0.5$, the value of $P_{reduction}$ is negative. Then the interest traffic of other interfaces will be automatically transferred to the current interface and forwarded according to the highest ranked interface.

Interest aggregation does not affect the effectiveness of PFECC. If the data packet marked as "reduce" reaches multiple downstream interfaces via PIT, this will reduce the sending window of each interface. This is exactly what we expect to see, as if the sending window of any downstream interface is not reduced, the congestion control mechanism proposed in this paper will not work.

Flows in PFECC can encounter multiple bottleneck links. In this case, the feedback is updated only when the congestion level detected by the router on the path is greater than the congestion level carried by the data packet, which ensures that the consumer uses the most serious feedback value calculated by the router on the path to determine the interest packets' sending rate. The specific method is that the feedback of each data packet at the producer is "add", and the router can change the feedback "add" to "reduce" according to the local link congestion, but it cannot be reversed. This ensures that each router can only add "reduce" marked packets, but not add "add" marked packets. As shown in Figure 3, C3 retrieves data from P2. Since there is no congestion on the path, C3 only receives packets marked as "add". C2 retrieves data from P1 and receives packets marked "reduce" from R2 because the link between R2 and R1 is congested. C1 also retrieves data from P1, but in addition to receiving packets with the congestion mark "reduce" from R2, C1 also receives from R1 because link R1 to C1 is also congested. This allows C1 to adapt to R1's bottleneck while C2 adapts to R2's bottleneck.

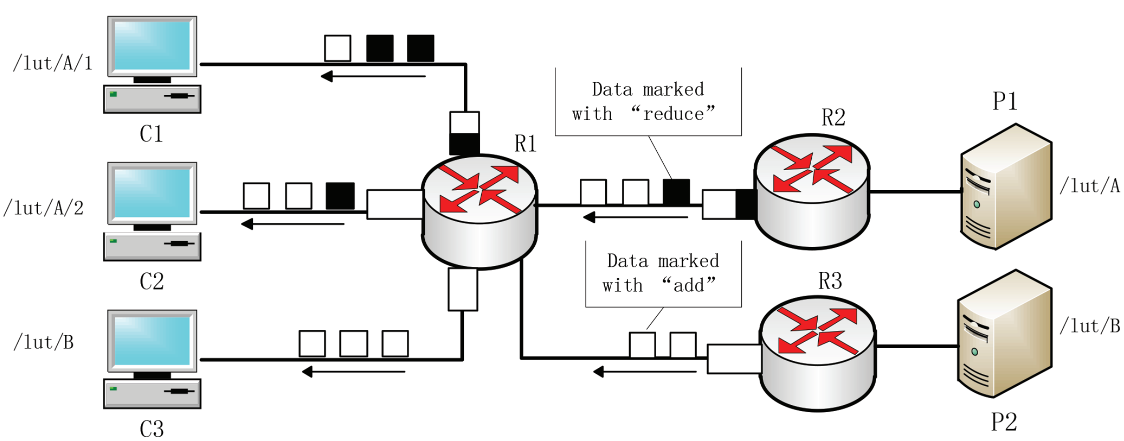

Figure 3. Multibottleneck links feedback.

4. Evaluation

We simulate PFECC under the ndnSIM2.6 platform and evaluate the performance of PFECC in various topological environments. The simulation parameters are set as follows: The parameter α in formula (1) is the target bandwidth utilization of the link, which is set to 0.95 in all experimental schemes in this paper. d_t is the value of the target delay, set to 5 ms. In the multipath forwarding scheme, the value of $MF\_minDelay$ reflects the sensitivity of the multipath algorithm to packet queuing delay, which is set to 3 ms.

4.1. The impact of interest flows

We first use the dumbbell topology (Figure 4) to evaluate the impact of the number of interest flows in formula (1). In this scenario, there are two producers and two consumers, and the bandwidth and delay are shown in the figure. Consumer1 retrieves Data from Producer1 and Consumer2 retrieves data from Producer2 (start later at 5 s). The results are shown in Figure 5. Without considering the flows of interest, when Consumer2 starts requesting data at 5 s, the queues will be established in Router1 and Router2. Consumer2 maintains an average length of 50 packets in Router2 before stopping at the 20 s. This is because if the influence of the number of interest flows is not taken into account, the target rate calculated in formula (1) will be larger, and then the number of packets marked with "add" will be increased, which will make the consumer send rate larger. This will cause a queue to appear in the router. This shows that it is necessary to consider the influence of the number of interest flows .

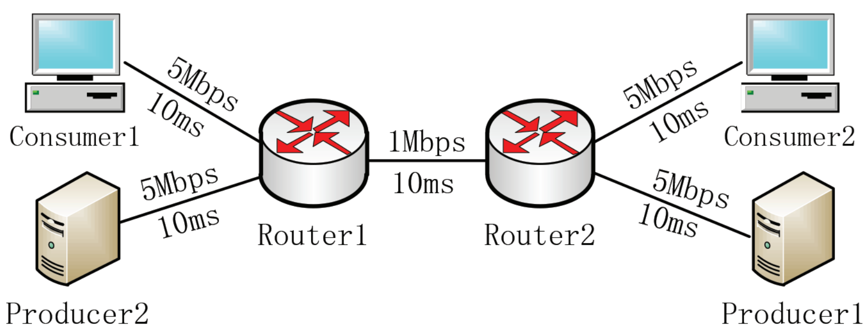

Figure 4. dumbbell topology.

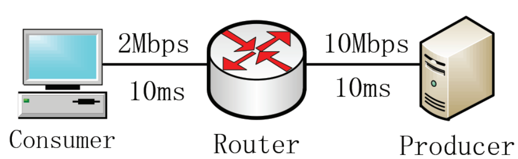

Figure 5. The impact of interest flows.

4.2. Three-node linear topology

We use the three-node linear topology in Figure 6 to evaluate the effect of the PFECC algorithm and the simple first input first output (FIFO) scheme. PFECC detects congestion in the queue of router interface and adds feedback on each data packet returned, so that the sending rate of consumers quickly converges to the target rate, it is essentially an AQM algorithm. In Figure 6, there is only one consumer, one router, and one producer, and no cache is used. Consumer of the FIFO scheme uses TCP's classic binary increase congestion control [20] conservative loss adaptive algorithm. We compare from the aspects of throughput, delay, and queue packet loss, respectively.

The results are shown in Figure 7. Simulation results show that the explicit marking scheme of the PFECC algorithm achieves almost the same throughput as the FIFO scheme, but in terms of packets transmission delay, the FIFO scheme using a drop-tail queue has great problems. The FIFO scheme always fills the queue completely first, which will cause the queue to be always too long, increasing the transmission delay of data packets and even the event of dropping. The advantage of the PFECC algorithm is that the queue reacts before it reaches its limit, thereby avoiding the queue being too long, and further avoiding packet loss and the increase of the transmission delay. Besides, increasing the queue size to more packets will not affect the performance of the PFECC algorithm, but will greatly increase the queuing delay of the FIFO scheme.

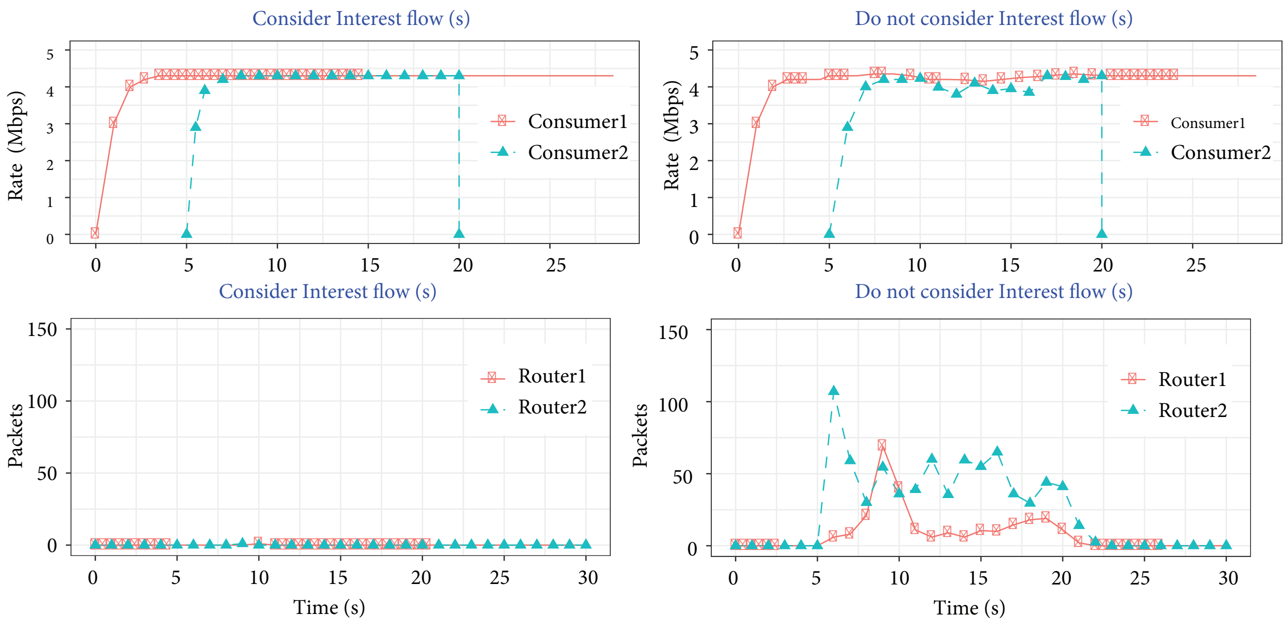

Figure 6. Dumbbell topology.

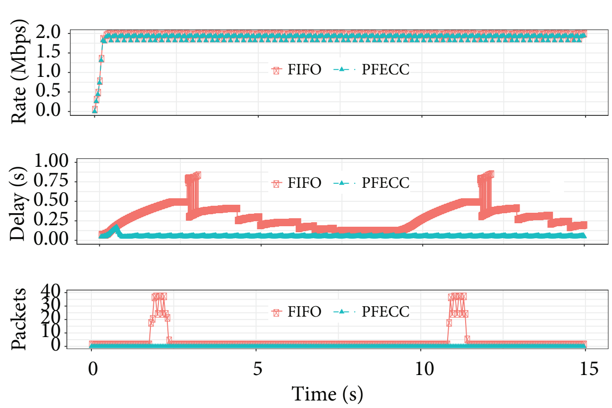

Figure 7. Consumer rate adjustment and AQM.

4.3. Multipath topology

We use the topology shown in Figure 8 to evaluate the throughput of the PFECC algorithm. In this topology, consumer i retrieves the content with the prefix /prefix/i, which is stored in all producers (P1, P2, P3, and P4). Consumers start retrieving content from different times (0 to 4 s). Simulation shows the throughput of all consumers of PFECC and their sum and compares the total throughput with PCON. PCON-1 means to use the shortest path priority forwarding strategy in PCON scheme, and PCON-2 means to use the forwarding strategy proposed by PCON. It can be seen from Figure 9 that PFECC can use the available bandwidth faster and maximize bandwidth utilization.

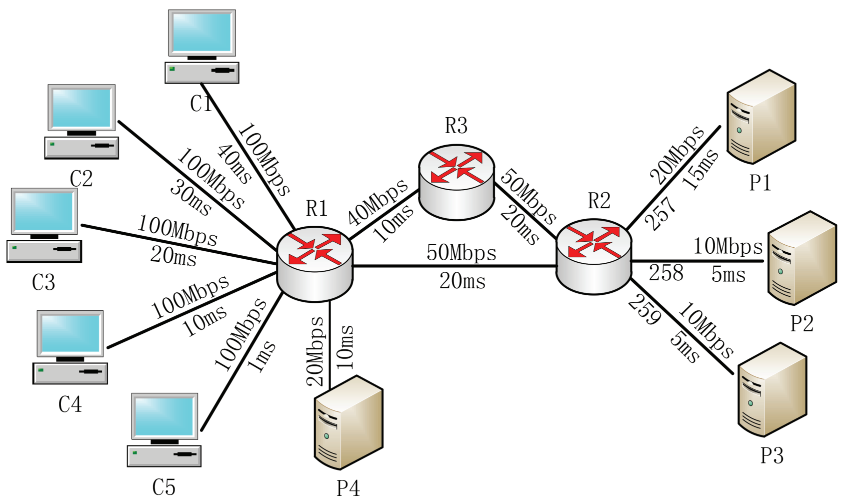

Figure 8. Multipath topology.

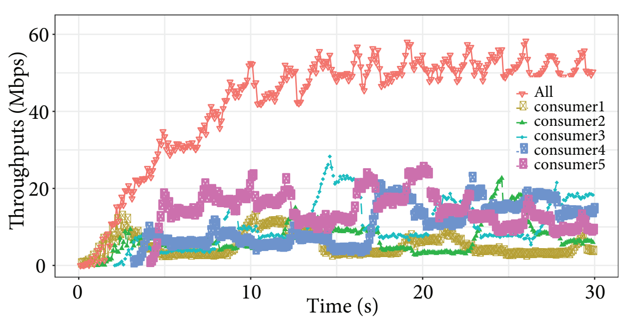

Figure 9. The throughput of PFECC.

The comparison of total throughput in Figure 10 shows that PFECC can use bandwidth faster than PCON and provides higher bandwidth utilization than PCON-1 and PCON-2. This is because the feedback algorithm that adds a mark to each data packet can more accurately and carefully adjust the size of the sending window of the consumers, and at the same time helps to enhance the stability of transmission. In addition, the PFECC multipath-aware adaptive forwarding strategy can split the forwarding ratio of interest in time before congestion becomes very serious, which avoids further congestion and improves throughput.

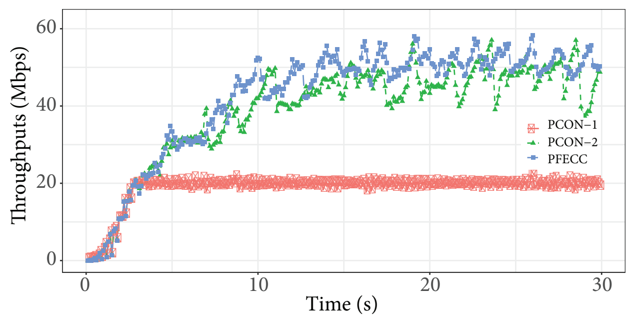

Figure 10. Throughput comparison between PFECC and PCON.

4.4. Performance of multipath-aware forwarding

To further evaluate the performance of multipath forwarding in PFECC, we used the topology composed of nodes R2, P1, P2 and P3 in Figure 8 and used three comparison schemes. In this topology, the consumer i retrieves content with the prefix / prefix / i, which is stored in all producers (P1, P2, P3, and P4). In this scheme, R2 acts as a consumer, and searches content from three producers (P1, P2, and P3). The three comparison schemes are as follows:

(1) Bandwidth and RTT are equal: The three paths R2-P1, R2-P2, and R2-P3 have equal 10 Mbps bandwidth and equal 10 ms RTT.

(2) Equal bandwidth and unequal RTT: The three paths have equal bandwidth of 10 Mbps, but RTTs are 10 ms (interface 257), 50 ms (interface 258), and 100 ms (interface 259).

(3) Unequal bandwidth and equal RTT: The three paths have equal RTT of 10 ms, but the bandwidths are 5 Mbps (interface 257), 15 Mbps (interface 258), and 40 Mbps (interface 259).

We compared PFECC with the other two multipath forwarding strategies, CF and PI, where PI selects the interface with the least Interest packets to be forwarded, and CF calculates the weight of each interface according to the number of PIs, and uses a weighted round-robin algorithm to select forwarding interface. After each algorithm runs stably, we calculate the forwarding ratio of each interface.

The results are shown in Table 2. All three forwarding strategies work well in the "equal" scheme, and the received traffic on three interfaces is almost equal. When RTT is different, PI and CF first choose the interface with lower RTT, which leads to a decrease in throughput. In a scenario with different bandwidths, PI and CF cannot use the optimal split ratio for interfaces with higher bandwidth. The expected optimal ratio is 8%:25%:67%, but they only reach about 17%:35%:48%, thereby reducing the overall throughput. In scenarios with different RTTs and different bandwidths, the results of PI and CF are consistent with the analysis by Nguyen in [21]. This analysis shows that the CF and PI forwarding strategies are biased against paths with longer RTT or higher bandwidth. Because the multipath forwarding strategy of PFECC directly responds to the congestion trend of upstream links, it can achieve the expected forwarding split ratio in all cases, which enables PFECC to achieve higher overall throughput faster.

4.5. Overall performance

We use the Abilene topology in Figure 11 to evaluate the overall performance of the PFECC algorithm according to flows completion time. In this topology, nodes 12, 16, and 19 are producers, and their content names are /alibaba, /tencent, and /baidu, respectively. Each dotted circle represents a node group, and the three

Table 2. The forwarding ratio of each interface in different schemes.

Algorithm	Split ratio of each interface								
	Equal			Diff_RTT			Diff_BW		
	257	258	259	257	258	259	257	258	259
PFECC(%)	33.0	34.0	33.0	33.5	33.0	33.5	8.0	25.0	67.0
CF(%)	33.5	33.0	33.5	50.0	28.0	22.0	17.0	35.0	48.0
PI(%)	33.0	34.0	33.0	72.0	18.0	10.0	16.0	36.0	48.0

consumers in the group request three different contents from the producers. The different colored arrows in the topology represent the forwarding path from the consumer to the content source. In the experiment, the content of the producer is of the same size and divided into 5000 content chunks with the size of 1 kB. The cache strategy uses the LRU strategy, and the size of the cache is 1000 packets. The requests of the consumers follow the Poisson distribution and start randomly between 0 and 5 s after the start of the experiment. We compare the flows completion time of the PFECC algorithm with the PCON algorithm. The result is shown in Figure 12. Compared with the PCON algorithm, PFECC can achieve a shorter total flow completion time in almost every node group. This shows that PFECC can improve the overall throughput of the network and provide users with better quality of service.

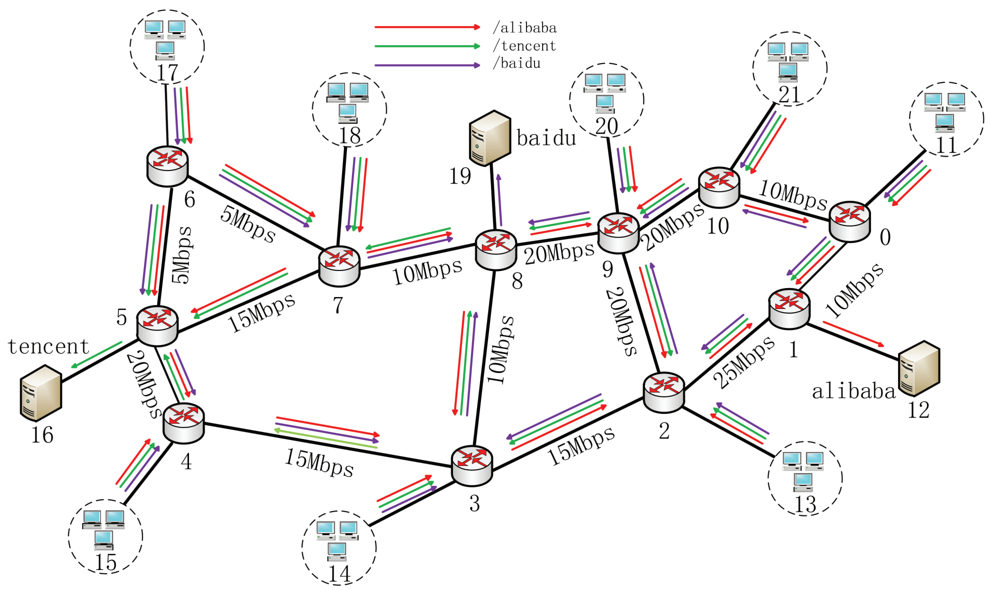

Figure 11. Abilene topology.

5. Conclusion

Inspired by the traditional TCP/IP network congestion control algorithm ABC, this paper proposes a congestion control algorithm PFECC in NDN. PFECC algorithm does not adopt the traditional method of detecting congestion based on RTO timeout or packet arrival interval, but detects congestion through the delay information in the queue and gives feedback on each data packet, which improves the accuracy of congestion detection and the timeliness of consumers' adjustment of sending rate. In the process of calculating feedback marks, PFECC

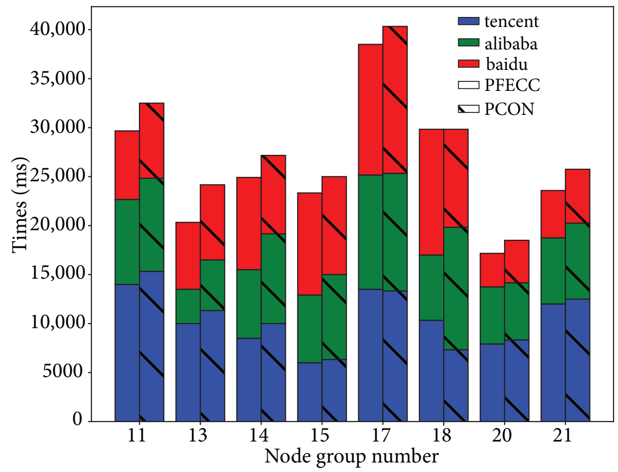

Figure 12. Flows completion time.

considers the influence of interest flows, distinguishes different flows at intermediate nodes, calculates the target rate, and maintains explicit feedback for each flow, achieving fairness between different flows. In addition, PFECC reuses the congestion field of the data packet for feedback. This method of data packet piggybacks is not only simple and easy to expand, but it will not further increase the load on the link. Finally, taking advantage of NDN's natural support for multipath forwarding, this paper proposes a multipath forwarding algorithm. The router splits the interest traffic according to the upstream link congestion trend. The proposed algorithm can react before the occurrence of congestion, which can avoid the occurrence of congestion effectively. When congestion is inevitable, consumers and intermediate nodes use different congestion feedback information to avoid excessive adjustment and improve network throughput. When the link congestion condition improves, the interest traffic will automatically switch to the best path for forwarding, which further improves the efficiency of adaptive forwarding. In future work, we will consider the bandwidth estimation method in PFECC and evaluate our scheme in more complex scenarios.

References

[1] Zhang L, Afanasyev A, Burke J, Jacobson V, Claffy K et al. Named data networking. ACM SIGCOMM Computer Communication Review 2014; 44 (3): 66-73. doi: 10.1145/2656877.2656887

[2] Floyd S, Jacobson V. Random early detection gateways for congestion avoidance. IEEE/ACM Transactions on Networking 1993; 1 (4): 397-413. doi: 10.1109/90.251892

[3] Nichols K, Jacobson V. Controlling queue delay. Communications of the ACM 2012; 55 (7): 42-50. doi: 10.1145/2209249.2209264

[4] Goyal P, Agarwal A, Netravali R, Alizadeh M, Balakrishnan H. ABC: a simple explicit congestion controller for wireless networks. In: 17th USENIX Symposium on Networked Systems Design and Implementation; Santa Clara, CA, USA; 2020. pp. 353-372.

[5] Mastorakis S, Afanasyev A, Moiseenko I, Zhang L. ndnSIM 2.0: A new version of the NDN simulator for NS-3. NDN, Technical Report NDN-0028, 2015.

[6] Carofiglio G, Gallo M, Muscariello L. ICP: Design and evaluation of an interest control protocol for content-centric networking. In: 2012 Proceedings IEEE INFOCOM Workshops; Orlando, FL, USA; 2012. pp. 304-309.

[7] Saino L, Cocora C, Pavlou G. Cctcp: A scalable receiver-driven congestion control protocol for content centric networking. In: 2013 IEEE International Conference on Communications (ICC); Budapest, Hungary; 2013. pp. 3775-3780.

[8] Carofiglio G, Gallo M, Muscariello L, Papali M. Multipath congestion control in content-centric networks. In: 2013 IEEE conference on computer communications workshops (INFOCOM WKSHPS); Turin, Italy; 2013. pp. 363-368.

[9] Carofiglio G, Gallo M, Muscariello L. Joint hop-by-hop and receiver-driven interest control protocol for content-centric networks. ACM SIGCOMM Computer Communication Review 2012; 42 (4): 491-496. doi: 10.1145/2377677.2377772

[10] Ren Y, Li J, Shi S, Li L, Chang x. An interest control protocol for named data networking based on explicit feedback. In: 2015 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS); Oakland, CA, USA; 2015. pp. 199-200.

[11] Zhang F, Zhang Y, Reznik A, Liu H, Qian C et al. A transport protocol for content-centric networking with explicit congestion control. In: 23rd international conference on computer communication and networks (ICCCN); Shanghai, China; 2014. pp. 1-8.

[12] Schneider K, Yi C, Zhang B, Zhang L. A practical congestion control scheme for named data networking. In: Proceedings of the 3rd ACM Conference on Information-Centric Networking; Kyoto, Japan; 2016. pp. 21-30.

[13] Rozhnova N, Fdida S. An effective hop-by-hop interest shaping mechanism for ccn communications. In: Proceedings IEEE INFOCOM Workshops; Orlando, FL, USA; 2012. pp. 322-327.

[14] Rozhnova N, Fdida S. An extended hop-by-hop interest shaping mechanism for content-centric networking. In: 2014 IEEE global communications conference; Austin, TX, USA; 2014. pp. 1-7.

[15] Wang Y, Rozhnova N, Narayanan A, Oran D, Rhee I. An improved hop-by-hop interest shaper for congestion control in named data networking. ACM SIGCOMM Computer Communication Review 2013; 43 (4): 55-60. doi: 10.1145/2534169.2491233

[16] Zhong S, Liu Y, Li J, Lei K. A rate-based multipath-aware congestion control mechanism in named data networking. In: 2017 IEEE international symposium on parallel and distributed processing with applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC); Guangzhou, China; 2017. pp. 174-181.

[17] Carofiglio G, Gallo M, Muscariello L. Optimal multipath congestion control and request forwarding in information-centric networks: Protocol design and experimentation. Computer Networks 2016; 110: 104-117. doi: 10.1016/j.comnet.2016.09.012

[18] Mahdian M, Arianfar S, Gibson J, Oran D. MIRCC: Multipath-aware icn rate-based congestion control. In: Proceedings of the 3rd ACM Conference on Information-Centric Networking; Kyoto, Japan; 2016. pp. 1-10.

[19] Bansal D, Balakrishnan H, Floyd S, Shenker S. Dynamic behavior of slowly-responsive congestion control algorithms. ACM SIGCOMM Computer Communication Review 2001; 31 (4): 263-274. doi: 10.1145/964723.383080

[20] Xu L, Harfoush K, Rhee I. Binary increase congestion control (BIC) for fast long-distance networks. In: IEEE INFOCOM 2004; Hong Kong, China; 2004. pp. 2514-2524.

[21] Nguyen D, Fukushima M, Sugiyama K, Tagami A. Efficient multipath forwarding and congestion control without route-labeling in CCN. In: 2015 IEEE international conference on communication workshop (ICCW); London, UK; 2015. pp. 1533-1538.