A Study on Performance of Explicit Rate Report based Congestion Control under Coarse-grained Clock Management

Toshihiko Kato, Takumi Enda, Ryo Yamamoto and Satoshi Ohzahata
Graduate School of Informatics and Engineering, University of Electro-Communications,
1-5-1, Chofugaoka, Chofu, Tokyo 182-8585 Japan

Keywords: NDN, Congestion Control, Rate Control, MIRCC, Clock Management.

Abstract: Named Data Networking (NDN) is a widely adopted future Internet architecture for large scale content retrieval. The congestion control in NDN is actively studied, and the rate-based congestion control method is considered to be well suited. From the viewpoint of implementation, however, the rate-based method has an issue that it requires the fine-grained clock management, which is hard to implement in off-the-shelf computers. Among the rate-based congestion control methods, an approach in which intermediate nodes report a maximum rate for a flow explicitly is considered to work well. In this paper we pick up MIRCC (Multipath-aware ICN Rate-based Congestion Control) as an example of explicit rate report based scheme, and examine the impact of system clock granularity of the performance of NDN rate based congestion control. This paper provides the performance evaluation when consumers and NDN routers use the system clock with long time interval.

1 INTRODUCTION

Resulting from a drastic increase in Internet traffic (Cisco, 2020), there are many studies on the future Internet architecture called Information Centric Network (ICN), which is well suited for large scale content retrieval. Named Data Networking (NDN) (Jacobson et al., 2009) is a widely adopted platform for the ICN researches. NDN relies on the name of required content, not the address of hosts containing the content, as a fundamental concept. NDN uses two types of packets in all communications: an *Interest packet* and a *Data packet*. A *consumer* that requests a specific content sends an Interest packet containing the content name. A *producer* that provides the corresponding content data returns a Data packet to the consumer. *NDN routers* transferring the Data packet cache the packet for future redistribution.

The congestion control is one of the hot research topics in NDN (Ren et al., 2016). Although it has been also a hot topic in TCP, the mechanisms in TCP congestion control are limited to the congestion window management at data senders (Afanasyev et al., 2010) and the simple explicit congestion notification at intermediate routers (Ramakrishnan et al., 2001). In contrast, various techniques can be introduced to the NDN congestion control. One is a receiver-driven window-based congestion control approach. In this approach, a consumer maintains a window size for Interest packet sending, just link a sender in TCP. Congestion is detected by timeout (Carofiglio et al., 2012) (Saino et al., 2013) or the congestion notification (Zhang et al., 2014), and the window for Interest packets are managed heuristically, e.g., through an Additive Increase and Multiplicative Decrease (AIMD) mechanism. Another methods adopts CUBIC TCP mechanism (Liu et al., 2016) or BIC TCP mechanism (Schneider et al., 2016) in the window control. In NDN, the rate-based congestion control approach is applicable. In this approach, a consumer and routers maintain a rate, in which Interest packets are transmitted contiguously. The rate is determined heuristically by use of congestion notification (Cheng et al., 2013) or by the explicit rate reporting (Rozhnova and Fdida, 2014) (Zhang et al., 2015) (Mahdian et al., 2016).

Among these methods, the rate-based method with explicit rate reporting provides the best performance. The overview of this method is as follows. Each router monitors the total of data packet traffic receiving over an individual link to an upstream router or a producer, and calculates the optimal Interest packet rate for each Interest-Data

Kato, T., Enda, T., Yamamoto, R. and Ohzahata, S.
A Study on Performance of Explicit Rate Report based Congestion Control under Coarse-grained Clock Management.
DOI: 10.5220/0009876200820088
In *Proceedings of the 17th International Joint Conference on e-Business and Telecommunications (ICETE 2020) - DCNET, OPTICS, SIGMAP and WINSYS*, pages 82-88
ISBN: 978-989-758-445-9

flow. Each router sends this optimal rate in a Data packet toward a consumer, and a consumer sends Interest packets according to the reported rate. Among the proposed explicit rate reporting methods, the Multipath-aware ICN Rate-based Congestion Control (MIRCC) specified in (Mahdian et al., 2016) provides the most stable and highest performance.

From the viewpoint of implementation, however, the rate-based congestion control approach has some problems. Since the transmission speed in recent data links becomes high, e.g., 10 Gbps in an access link, the fine-grained clock management is required in the rate-based congestion control. For example, if the Data packet size is 10,000 bits and the link speed is 10 Gbps, the interval of Interest packet transmission is 1 micro seconds (corresponding to 1 MHz) when Interest packets are transmitted in a line speed. If the rate is 5 Gbps or 3 Gbps, the Interest transmission interval will be 2 micro seconds (500 KHz) or 3.33 micro seconds (300 KHz), respectively. In order to handle these cases, it is supposed that higher precision clock with shorter tick, such as 0.1 micro second (10 MHz), will be required to control the Interest packet sending timing.

On the other hand, it is considered that the fine-grained clock management is hard to implement in off-the-shelf computers. For example, TCP implementation uses 200 msec (5Hz) and 500 msec (2Hz) clocks for the delayed acknowledgement and retransmission, respectively (Fall and Stevens, 1994). Therefore, implementing rate-based mechanism with micro second order clock is extremely hard.

We pointed out this issue and discussed how a coarse-grained clock system influences the NDN rate-based congestion control, in our previous paper (Kato et al., 2018). We adopted the Stateful Forwarding (Cheng et al., 2013) as a target system of the evaluation, and showed that the performance, specifically the Data packet throughput, is degraded largely when a coarse-grained clock is introduced. Moreover, we proposed a method to send Interest packets more smoothly even in the coarse-grained clock environment.

However, the Stateful Forwarding is not the best example of the NDN rate-based congestion control methods. As stated above, the explicit rate reporting methods, especially MIRCC, provide better performance. In this paper, we examine how the coarse-grained clock system influence the performance of MIRCC.

The rest of this paper is organized as follows. Section 2 gives the related work focusing on the overview of NDN congestion control and MIRCC. Section 3 describes the implementation of MIRCC over the ndnSIM simulator (Afanasyev et al., 2012), which is a widely used network simulator for NDN, and the modification for the coarse-grained clock system. Section 4 gives the performance evaluation results. In the end, Section 5 concludes this paper.

2 RELATED WORK

2.1 Related Work on NDN Congestion Control

As described above, the congestion control methods in NDN are categorized as the window-based and the rate-based methods. As for the window-based methods, we can pick up the following three examples. The Interest Control Protocol (ICP) (Carofiglio et al., 2012) and the Content Centric TCP (CCTCP) (Saino et al., 2013) are examples of the traditional TCP like window-based methods. Here, a consumer sends Interest packets continuously within the window size, and the window size is changed according to the AIMD mechanism triggered by Data packet reception and congestion detected by timeout. The Chunk-switched Hop Pull Control Protocol (CHoPCoP) (Zhang et al., 2014) is another window-based method. It introduces the explicit congestion notification with random early marking instead of the timeout-based congestion detection. Although these use the AIMD mechanism, (Liu et al., 2016) proposes a window control according to CUBIC TCP. The Practical Congestion Control (PCON) scheme (Schneider et al., 2016) uses the CoDel active queue management scheme (Nichols and Jacobson, 2012), which watches out the delay of packets in sending queues, to detecting congestion. When congestion is detected, a router signals it to consumers by explicitly marking Data packets. In respond to this reporting, the rate reducing is performed by consumers. PCON follows AIMD or BIC TCP for the window control. In the window-based methods, the window size itself may not be optimal resulting from the Data packet caching in different routers.

On the other hand, the rate-based methods are classified into the non-deterministic scheme, which uses the AIMD mechanism in determining the Interest sending rate, and the explicit rate notification scheme, in which intermediate routers report the optimal Interest rate to a consumer. The Stateful Forwarding (Cheng et al., 2013) is an example of the former scheme. It introduces a negative acknowledgment (NACK) packet as a response to an Interest packet. It is sent back to a consumer when an Interest packet is discarded in an intermediate router.

A consumer and a router manage the Interest sending rate locally by AIMD, when a Data packet is received, or a NACK packet is received / a timeout happens.

In contrast with those non-deterministic methods, new methods have been introduced that enable routers to report a maximum allowed Interest sending rate. In the Hop-By-Hop Interest Shaping (HoBHIS) (Rozhnova and Fdida, 2014), routers decide the maximum allowed Interest sending rate independently and accordingly shape Interest packet. The maximum allowed rate is also reported to a consumer and this allow a consumer to send Interest packets without invoking congestion. In the Explicit Congestion Notification (ECN) based Interest sending rate control method proposed in (Zhang et al., 2015), a consumer uses a minimum rate among the reported rates from all intermediate routers. MIRCC introduces a similar per-link Interest shaper at every router and rate reporting to consumer. It takes account of the case that a flow uses multipath transfer.

In those methods, the maximum allowed rate is calculated from the parameters including link capacity and utilization, queue size, inflated Interest rate, and average round-trip time (RTT). They are able to control Interest transmission so as to suppress congestion, and as a result they can provide higher throughput compared with other rate-based methods.

In spite of these merits, their implementation requires the precise timing control for sending Interest packets. However, the fine-grained clock is hard to implement. There are some traditional rate-based schemes, but they use some hardware mechanism instead of the fine-grained clock. For example, the Asynchronous Transfer Mode (ATM) uses a kind of rate-based cell transfer (ITU-T, 1999), but ATM uses null cells discarded at a receiving side to regulate cell streams at a specific rate. (Yamamoto, 2008) introduced pause packets over Gigabit Ethernet, corresponding to null cells in ATM, that are used only between end nodes and switching hubs. Those approaches are implemented by the MAC level hardware, but NDN Interest packets are handled as a higher level, which makes the hardware support difficult.

2.2 Details of MIRCC

In MIRCC, consumers and routers that forward Interest packets, called forwarders, maintain the parameters indicated in Table 1. Each forwarder calculates the Interest sending rate *R(t)* for individual flows, at each interval *T*. *R(t)* is specified as the sum of *base_rate(t)* and *excess_rate(t)*, each of which is given in the following way.

Table 1: List of MIRCC parameters.

Parameter	Definition
R(t)	Stamping rate at time *t*
C	Capacity of upstream link
N	Equivalent number of flows with full rate
T	Interval of rate calculation
q(t)	Inflated instantaneous queue size
y(t)	Incoming Interest rate during [*t*-*T*, *t*)
d(t)	Smoothed average RTT
β (t)	Self-tuned parameter for stability
η	Target link utilization

In order to calculate *base_rate(t)*, a forwarder estimates the number of flows by equation (1).

$$N = {max(C, y(t))}/{R(t-T)} \tag{1}$$

Then, *base_rate(t)* is computed as follows:

$$base\_rate(t) = \frac{\eta C - \beta(t)\frac{q(t)}{d(t)}}{N} \tag{2}$$

Here, $\beta(t)$ is given by

$$\beta(t) = max\left(0.1, \frac{y(t)-y(t-T)}{y(t)}\right) \tag{3}$$

As for *excess_rate(t)*, the following equation is used.

$$excess\_rate(t) = R(t-T) - {y(t)}/{N} \tag{4}$$

In order to avoid high-frequency oscillation, an exponential weighted moving average (EWMA) is applied to both *base_rate(t)* and *excess_rate(t)* with weight 0.5. Finally, *R(t)* is given by the following equation.

$$R(t) = base\_rate\_ewma(t) + excess\_rate\_ewma(t) \tag{5}$$

When a router receives a Data packet, it compares the stamping rate included in the packet, and if the included rate is larger than the computed *R(t)*, then *R(t)* is set in the Data packet.

3 MIRCC WITH COARSE-GRAINED CLOCK

3.1 Implementation of MIRCC over ndnSIM

In order to evaluate the performance of MIRCC described in the previous section, we implemented it over the ndnSIM simulator version 1.0. The reason we used this version is that we reused the coarse-gained clock implementation in our previous paper. We implemented MIRCC in the following way.

(1) Add *R(t)* Parameter in Data Packet

In order to convey *R(t)* in a Data packet, we defined the corresponding parameter, `m_rate`, and the methods to access and modify it, in files `model/ndn-data.{h,cc}`. Besides, the methods for formatting a Data packet, `Serialize()` and `Deserialize()`, is modified in `model/wire/ndnsim.cc`.

(2) Implement a Method Calculating *R(t)*

A method called `CalculateRate()` is implement-ed in `utils/ndn-limits.cc`. This method is invoked every T interval, and calculates *R(t)* according to equations (1) through (5) specified above. Here, we need to mention that *y(t)* is given by dividing the number of received Interest packets by T, that the leaky bucket size is used as *g(t)*, and that η=1 in this case.

(3) Add Various Functions in Interest/Data Handling

Interest and Data packets are handled in file `model/fw/ndn-forwarding-strategy.cc`. We added the following functions in this file.

- Counting received Interest packets.
- Evaluating smoothed RTT at receiving Data packets.
- Setting *R(t)* in a Data packet if the corresponding value in the Data packet is larger than the calculated *R(t)*.

(4) Behaviour of Consumer

A consumer sends Interest packets according to the reported *R(t)* in Data packets. We implemented this kind of consumer as a new class called `ConsumerLi`.

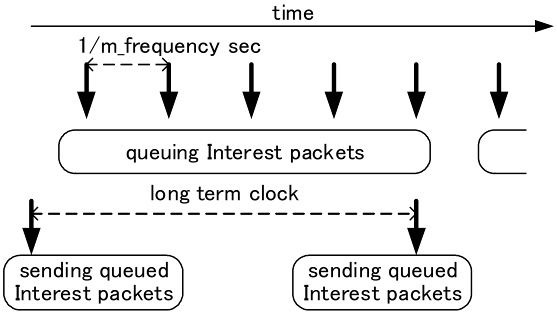


Figure 1: Implementation scheme of coarse-grained clock system in a consumer.

3.2 Implementation of Coarse-grained Clock based MIRCC

In the original NDN, the rate control in the consumer is implemented as follows. The sending of Interest packets with a specific rate is done in the `ScheduleNextPacket()` method of the `ConsumerLi` class. In this method, the `SendPacket()` method of the `Consumer` class, which is the superclass, is invoked periodically, every `1.0/m_frequency` seconds. The `SendPacket()` method sends one Interest packet actually.

We emulated a course-grained clock system in the `Consumer` class in the following way (see Figure 1).

- A clock system with longer tick, such as 100 msec, is implemented in the `ConsumerLi` class. It calls itself periodically with the `Schedule()` method of the `Simulator` class.
- We also introduced a queue storing Interest packets temporarily. This queue is implemented using the list class.
- In the `SendPacket()` method, Interest packets are stored in the queue, instead of being sent actually.
- When the longer clock tick is invoked, all the queued Interest packets are transmitted actually.

In the router side in MIRCC, we implemented a coarse-grained clock system, by assigning a large value, such as 100 msec, in the interval T. Basically, we assumed that the longer tick in a consumer and T in a router take the same value. We also evaluate the cases when the longer tick and T take different values.

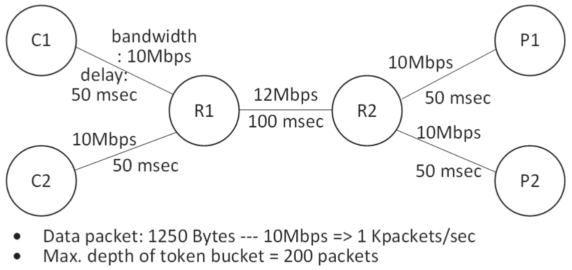


Figure 2: Network configuration for performance evaluation.

4 EXPERIMENT RESULTS

4.1 Experimental Setup

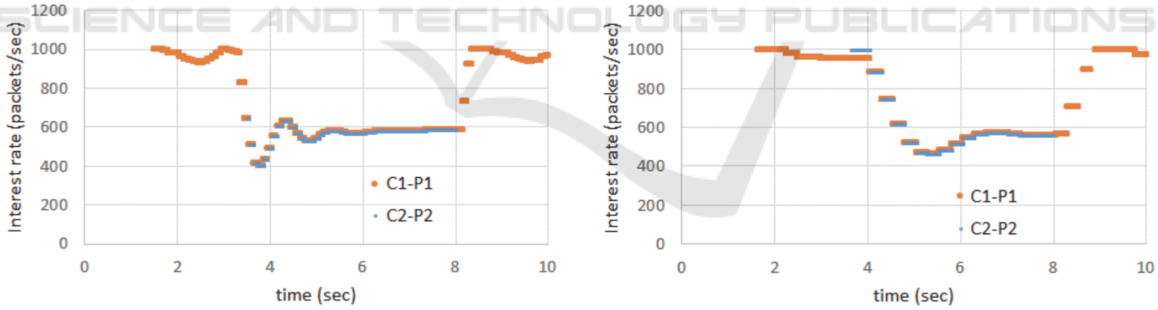
The network configuration used in this evaluation is shown in Figure 2, which is a dumbbell configuration where two consumers (C1 and C2) and two producers (P1 and P2) are connected through two routers (R1 and R2). The bandwidth and delay between a consumer and a router, and between a router and a producer are 10Mbps and 50 msec, respectively. Those between routers are 12Mbps and 100 msec, respectively. The length of a Data packet is 1250 bytes, and so the link speed 10Mbps and 12Mbps corresponds 1,000 packets/sec and 1,200 packets/sec, respectively. The depth of a token bucket used for policing the Interest packet flow is set to 200 packets in consumers and routers.

Under these conditions, we evaluated the cases that the coarse-grained clock has 100 msec, 250 msec, 500 msec and 750 msec interval values. In all the evaluation runs, consumer C1 transmits Interest packets to producer P1 between time 1 sec and 10 sec, and consumer C2 sends Interest packets to producer P2 between time 3 sec and 8 sec. In this evaluation, cache is not used.

4.2 Results of Performance Evaluation

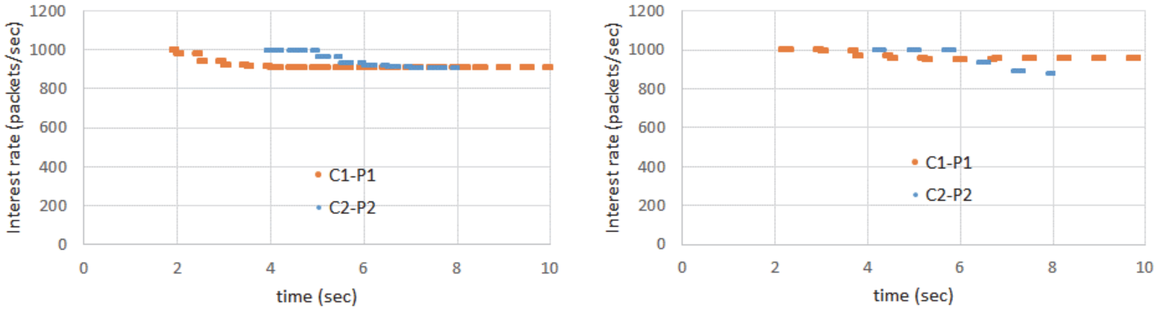
First of all, we evaluated the performance when the longer tick in consumers and interval *T* in routers have the same value. Figure 3 shows the results, where the time interval value is set to 100 msec, 250 msec, 500 msec, and 750 msec. The graphs show the Interest sending rate at consumers C1 and C2.

The result in Figure 3(a) corresponds to the case that the time interval is 100 msec. In the beginning, the Interest sending rate at C1 takes the value around 1,000 packets/sec although there is little fluctuations. When C2 starts the communication the Interest sending rates at C1 and C2 go to 600 packets/sec, with some fluctuations. This result shows that two flows from C1 and C2 share the bandwidth of the bottleneck link evenly. It is considered that MIRCC performs well, even if the time interval is relatively large.



(a) time interval = 100 msec.

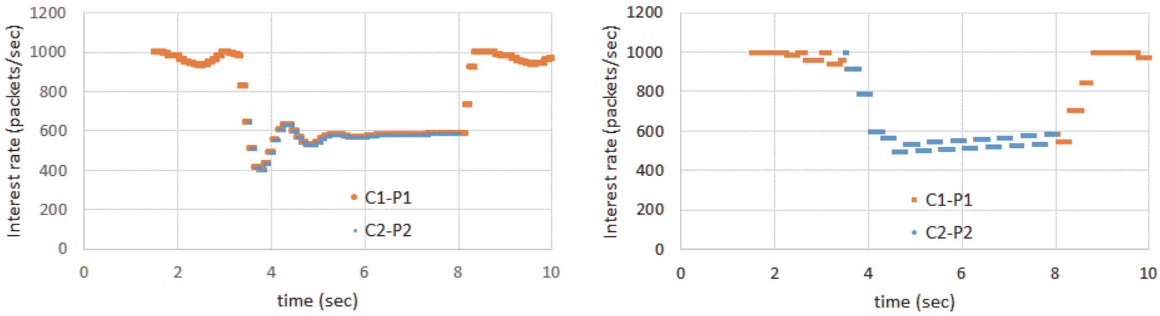
(b) time interval = 250 msec.



(c) time interval = 500 msec.

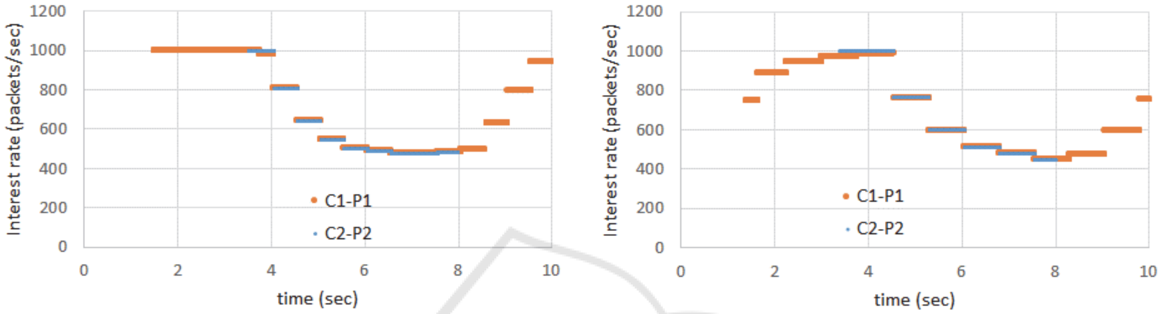
(d) time interval = 750 msec.

Figure 3: Results when longer tick and interval *T* are equal.



(a) interval T = 100 msec.

(b) interval T = 250 msec.



(c) interval T = 500 msec.

(d) interval T = 750 msec.

Figure 4: Results when longer tick is set to 100 msec.

Figure 3(b) shows the result in the case that the time interval is 250 msec. The result seems to be similar with the case of 100 msec. The difference is the convergence to 600 packets/sec when C2 starts the communication takes larger period, around 2 sec. In contrast these results, those in Figures 3(c) and 3(d) show a catastrophic situation. Even if C2 starts its session, the Interest sending rates in C1 and C2 keep 1,000 packets/sec as if the individual consumer communicates alone. These results indicate that the MIRCC mechanism does not work well. It should be noted that the time interval values in those cases are larger than RTT (400 msec) between the consumer and the producer.

Next, we evaluated the performance when the longer tick in consumers is set to 100 msec and interval T in routes has large values. Figure 4(a) is the result when T is 100 msec, and this is the same as the result shown in Figure 3(a).

Figure 4(b) is the result when interval T is 250 msec. Comparing with the result in Figure 3(b), the tendency of Interest sending rate is similar. In both cases, the rates in two consumers are set to 600 packets/sec while C2 communicates with P2.

The results in Figures 4(c) and 4(d) are completely different from those in Figures 3(c) and 3(d). When the longer tick in consumers is set to 100 msec, the MIRCC mechanism works well even if interval T in routers is set to 500 msec or 750 msec.

5 CONCLUSIONS

In this paper, we showed the performance evaluation of MIRCC, which is an effective rate-based congestion control mechanism, under the condition that the coarse-grained clock system is used in consumers and routers. The congestion control mechanism is an important issue in NDN, and the MIRCC is considered as one of the effective approaches. However, MIRCC requires a precise system clock and it is an implementation issue for off-the-shelf computers. We evaluated how the Interest sending rate is realized when the coarse-grained system clock is used in consumers and routers. The evaluation results showed that a large clock interval degrades the performance of MIRCC. The results also showed that the clock granularity in consumers for controlling Interest sending is more important than the control interval used by MIRCC in routers. Even if the interval values in routers is large, the performance of MIRCC is kept well when the system clock interval value is small in consumers.

REFERENCES

Cisco public White paper, 2020. *Cisco Annual Internet Report (2018-2023)*. https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.pdf.

Jacobson, V., et al., 2009. Networking Named Content. In *Proc. of CoNEXT '09*, *pp. 1-12*.

Ren, Y., Li, J., Shi, S., Li, L., Wang, G., Zhang, B., 2016. Congestion control in named data networking - A survey. *Computer Communications*, *vol. 86*, *no. 15*, *pp. 1-11*.

Afanasyev, A., Tilley, N., Reiher, P., Kleinrock, L., 2010. Host-to-Host Congestion Control for TCP. *IEEE Commun. Surveys & Tutorials, vol. 12, no. 3, pp. 304-342*.

Ramakrishnan, K., Floyd, S., Black, D., 2001. *The Addition of Explicit Congestion Notification (ECN) to IP*. IETF RFC 3168.

Carofiglio, G., Gallo, M., Muscariello, L., 2012. ICP: Design and Evaluation of an Interest Control Protocol for Content-Centric Networking. In *Proc. of IEEE INFOCOM 2012*, *pp. 304-309*.

Saino, L., Cocora, C., Pavlou, G., 2013. CCTCP: A Scalable Receiver-driven Congestion Control Protocol for Content Centric Networking. In *Proc. of IEEE ICC 2013*, *pp. 3775-3780*.

Zhang, F., Zhang, Y., Reznik, A., Liu, H., Qian, C., Xu, C., 2014, A Transport Protocol for Content-Centric Networking with Explicit Congestion Control. *In Proc. of IEEE ICCCN 2014*, *pp. 1-8*.

Liu, Y., Piao, X., Hou, C., Lei, K., 2016. A CUBIC-Based Explicit Congestion Control Mechanism in Named Data Networking.*In Proc. of IEEE CyberC 2016*, *pp. 360-363*.

Schneider, K., Yi, C., Zhang, B., Zhang, L., 2016. A Practical Congestion Control Scheme for Named Data Networking. In *Proc. of ACM ICN 2016*, *pp. 21-30*.

Cheng, Y., Afanasyev, A., Moiseenko, I., Zhang, B., Wang, L., Zhang, L., 2013. A case for stateful forwarding plane, *Computer Communications*, *vol. 36*, *no. 7*, *pp. 779-791*.

Rozhnova, N and Fdida, S., 2014. An extended Hop-by-Hop Interest shaping mechanism for Content-Centric Networking. In *Proc. of IEEE GLOBECOM 2014*, *pp. 1-7*.

Zhang, J., Wu, Q., Li, Z., Kaafar, M., Xie, G., 2015. A Proactive Transport Mechanism with Explicit Congestion Notification for NDN. In *Proc. of IEEE ICC 2015*, *pp. 5242-5247*.

Mahdian, M., Arianfar, S., Gibson, J., Oran, D., 2016. Multipath-aware ICN Rate-based Congestion Control. In *Proc. of ACM ICN 2016*, *pp. 1-10*.

Fall, K. and Stevens, W., 1994. *TCP/IP Illustrated, Volume1; The Protocols, Second Edition*. Addison-Wesley.

Kato, T., Osada, K., Yamamoto, R., Ohzahata, S., 2018. A Study on How Coarse-grained Clock System Influences NDN Rate-based Congestion Control. In *Proc. of IARIA ICN 2018*, *pp. 35-40*.

Afanasyev, A., Moiseenko, I., Zhang, L., 2012. *ndnSIM: NDN simulator for NS-3*. NDN, Technical Report NDN-0005.

Nichols, K. and V. Jacobson, V., 2012. Controlling Queue Delay. *ACM Magazine Queue*, *vol. 10*, *issue 5*, *pp. 1-15*.

ITU-T, 1999. *B-ISDN asynchronous transfer mode functional characteristics, Series I: Integrated Services Digital Nework, Recommendation I.150*.

Yamamoto, Y., 2008. Estimation of the advanced TCP/IP algorithms for long sistance collaboration. *Fusion Engineering and Design*, *vol. 83*, *issue 2-3*, *pp. 516-519*.

NDN, *Overall ndnSIM documentation; Forwarding Strategies.* http://ndnsim.net/1.0/fw.html.