Consumer-driven Adaptive Rate Control for Real-time Video Streaming in Named Data Networking

Takahiro Yoneda † Ryota Ohnishi † Eiichi Muramoto † Jeff Burke ‡

† Advanced Research Division, Panasonic Corporation, Kadoma City, Osaka, 571-8501 Japan.

‡ School of Theater, Film and Television, University of California, Los Angeles, CA 90095-1622 USA.

E-mail: {yoneda.takahiro, ohnishi.ryota muramoto.eiichi}@jp.panasonic.com, jburke@remap.ucla.edu

Abstract Named Data Networking (NDN) is a new networking architecture. This paper focuses on transport control for real-time video streaming in NDN. There have been several studies that address reliability and congestion control issues in the NDN transport mechanism area. In contrast, little attention has been paid to real-time transmission for video streaming. To improve the performance of real-time video streaming scheme on NDN, we propose a consumer-driven adaptive rate control mechanism. Our mechanism would provide scalability, good throughput, and low-latency transmission, without increasing the queuing delay and packet loss in the network. Our proposal also considers RTT (Round-Trip Time) fair data transmission, which enables several consumer nodes to fairly share network bandwidth of a bottleneck link regardless of the difference of network delay between publisher nodes and consumer nodes. This consideration is very important for networks in which the content transmitter dynamically varies. To identify the effectiveness of the proposal, we evaluate the performance of our mechanism with ns-3. The simulation results show that the proposed scheme is effective for real-time video streaming in NDN.

Keyword NDN; CCN; ICN; Real-time video streaming; Rate control; Fairness

1. Introduction

Named Data Networking (NDN) [1] is a new networking paradigm which was originally proposed as Content Centric Networking (CCN) by Van Jacobson et al. [2]. In NDN, all nodes communicate with Content Names that identify Content (data) itself. Basically, there are two types of messages: Interest packets and Data packets. A consumer node which is requesting Content sends an Interest packet with the Content Name; a publisher node generates the Content and sends Data packets, which include the Content Name and the Content. Additionally, the Pending Interest Table (PIT) and Content Store mechanism of an NDN router permits efficient packet transmission for the same request and data. As a future Internet architecture, NDN is expected to solve problems related to the current Internet's increasing traffic. However, transfer speed and memory capacity of the router are far from enough to handle the traffic explosions caused by increasing numbers of devices connected to the network. If the number of Interest packets which do not match with the Content Store of a router, increase, Data packets beyond the capacity flow into the network link. As a result, queuing delay and packet loss naturally increase. In addition to the recent trend of increased the number of devices, network traffic is also shifting from wired connections to wireless connections. And on these wireless networks, such as cellular networks and WLAN (Wireless Local Area Network), wireless network resources are more limited; wireless link capacity is dynamically changed. Therefore, a scalable and adaptive transport mechanism is an important topic for NDN. Fair share of the resources in best effort networking, as often discussed as it relates to conventional Internet architecture, is also an important topic for NDN architecture design. However, because of NDN's unique characteristics in the data transport mechanism, we cannot take an approach similar to the conventional Internet to achieve fairness. For example, the source of a Data packet may change regularly between publisher node and caches on routers located on the forwarding path of Interest packets. Additionally, NDN's multi-source nature allows the existence of a plural network path to one Data packet. In the case of conventional Internet architecture, most fair share discussions are based on the system model where all clients share the same single network path to each server. In case of NDN, we think it is insufficient to discuss the throughput fairness as it relates to the style of TCP (Transport Control Protocol)-related research [3,4], which assumes that the senders and receivers sharing a bottleneck link have the same network delay.

Meanwhile, video traffic accounted for half or more of all Internet traffic in 2012, and it is expected that video traffic will increase in the future [5]. It is also expected that video applications which require real-time transmission via the network, such as peer-to-peer video calls, multi-point video conferences and remote lectures [6], will become even more popular. The same may be said of NDN; real-time video transportation presents unavoidable challenges.

In order to maintain both low-latency and high video quality, adaptive rate control, according to variable link capacity and the traffic condition of the link, is necessary. In this paper, we propose an adaptive rate control mechanism especially focused on real-time video streaming in NDN. Furthermore, we discuss fair share of the resources of NDN where the network path is dynamically changed.

The remainder of the paper is organized as follows: Section 2 describes related work. Section 3 identifies challenges for real-time video streaming in NDN, and describes our proposed real-time transmission control mechanism for video streaming. Next, our evaluation with the network simulator is written in Section 4. Finally, Section 5 presents a conclusion.

2. Related works

To date, there are two types of approaches to congestion control mechanisms in NDN/CCN or Information Centric Network (ICN) research. One is a consumer-driven end-to-end approach. The other is the router playing certain role in the congestion control procedures.

[7-9] are pure consumer-driven congestion control mechanisms. These mechanisms, which are based on window-based rate control like AIMD (Additive Increase/Multiplicative Decrease) [10], has advantage in congestion avoidance and adaptability for network condition. However, it does not decrease the bit-rate until the consumer node detects any packet loss or re-transmission time-out, so they keep expending network bandwidth. These approaches sometimes significantly increase the queuing delay in the router. This indicates congestion trends and badly affects interactive real-time video streaming which essentially requires low-delay transmission. Furthermore these mechanisms assume that competing Data and Interest flows are sharing the same network path to maintain throughput fairness in a bottleneck link. For NDN transport mechanism, we think it is insufficient to discuss the throughput fairness, as described in the previous section.

[11] takes the hop-by-hop intermediate node support approach in [7], which defines the virtual queue for Interest packet flow in the router, and controls the maximum size of each virtual queue to satisfy the fairness of Data throughput among several competing consumer nodes. This mechanism has problems in its deployment among the existing routers, as well as a scalability problem in resource consumption which is proportionate to the number of flows.

In ICN, another criterion of fairness exists, such as the memory consumption used by the Content Store in the router [12-14]. But this criterion deeply depends on the cache management algorithm on the router, and it is not our research target.

There are several works that deal with video distribution of ICN [15-19]. In terms of video transmission, [15,16] employ window-based flow control with considering live video distribution. However, the window size is fixed in their implementations, and then adaptive video rate control and fairness are not duly considered. [17-19], which is based on MPEG-DASH [20], control requesting video streams adaptively. These works do not consider real-time transmission and fairness.

3. Design

This section describes our proposed mechanism for real-time video streaming.

3.1. Target Application

In order to clarify our target, we describe applicable application for proposed mechanism. In this paper, we assume video conferencing and interactive remote lecture system, which require real-time video distribution for interaction.

In these application, a publisher node divides each video frame into serial segments, and each segment is given a prefix name and sequential number as a Content Name, such as NDNVideo [16]. Each segment size is decided by each application. We assume that a consumer node can get the Content Name of latest video frame, before starting to receive the video stream. For instance, NDNVideo shows a method to get a sequential number of latest video frame with the Interest packet, by setting “ChildSelector” to “RIGHTMOST” [21]. The consumer node continually sends Interest packets with above Content Name while increasing the sequential number in order to get the whole segments. In the network, these Interest packets are forwarded toward the publisher node or the NDN router which has the segments in Content Store. The segments are forwarded as a Data packet to the consumer node from the publisher node or the NDN router.

3.2. Objectives

Low-latency transmission

For interactive video applications which require real-time video data transmission, such as video conferences and remote lectures, network latency increases are a fatal problem. Therefore, it is necessary to avoid queuing delay increases and packet loss in the router in order to maintain the real-time-ness of network transmission.

Keeping available best throughput

For interactive video applications, it is also important to keep good throughput during transmission. In the case of these applications, the network throughput directly affects the video quality. Basically, the video encode rate has to be under the available (idle) network bandwidth in order to keep low-latency transmission. Therefore, the capability to adapt to the actual available bandwidth is crucial.

Considering source variation

For the NDN transport mechanism, it has to consider that the RTT (Round-Trip Time) between sending an Interest packet and receiving the corresponding Data packet might be changed, because of the symptom of the congestion or the variation of the source (Data packet transmitter). Especially in the case of real-time applications which provide live content, a plurality of consumer nodes tends to access the latest content data simultaneously. In such a situation, these consumer nodes transmit the Interest packets for same content data at the same time. In the concrete, Fig. 1 shows a typical example of this phenomenon. Each consumer node (C1, C2 and C3) shares the same links connecting them to the publisher node. With these circumstances, when each consumer node independently sends a series of Interest packets at the same time, some of these Interest packets duplicate others in flight. Then, the routers can send the Data packets with Content Store or can suppress forwarding the same Interest packets by aggregating them within the PIT (if it does not answer from the Content Store). As a result, the location of the source varies from the publisher to the routers. For instance, when C3 sends Interest packets later than other consumer nodes, the Interest packets from C3 match the Content Store on Router2, and Router2 responds to C3 (see Fig. 1(a)). In other cases, when C1 sends Interest packets later than C2, the Interest packets from C1 match the Content Store on Router1 and Router1 responds to C1 (see Fig. 1(b)). In this case, the RTT which is measured on C1 is shorter than Fig. 1(a). In another case, when C2 stops sending Interest packets, the Interest packet from C3 is forwarded to Router1, and Router1 sends Data packets to C3 (see Fig.1(c)). In this case, the RTT which is measured on C3 is longer than other cases. In this way, the RTT measured on the consumer node may vary independently from the indication of congestion. In case of the real-time video streaming, the transport mechanism on NDN is required to provide congestion control algorithms which can distinguish whether the network is congested or not.

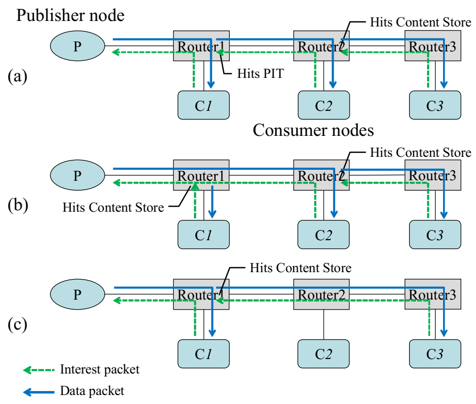

Fig. 1 Example of the source variation

Maintain fairness

It is preferable to share the competing network resources fairly between each content receiver using the common network resources at the same time [22-24]. In NDN, the location of the source dynamically varies because of the router mechanism of the Content Store and PIT, as described in the previous section. As stated, we think it is insufficient to discuss only about the throughput fairness in the style of TCP-related research, which assumes that senders and receivers sharing a bottleneck link have the same network delay.

We focus on the fairness of the bandwidth consumption in the shared bottleneck link located between publisher node and the consumer node. This fairness should include being tolerant about delays related to the network path (hereinafter referred to as RTT Fairness).

Scalability

Numbers of devices which connect to the network will increase in the future, and this will increase traffic. A scalable transport mechanism is an important topic – for a router with state information such as the PIT or Content Store.

3.3. Basic design

Consumer-driven

The consumer-driven approach was chosen for our proposed transport mechanism for real-time video streaming. Each consumer node autonomously controls the sending rate of the Interest packets, according to the receiving result of the Data packet corresponding to the Interest packet sent before. We assume that this rate control mechanism will be implemented inside an application of the consumer node, the Interest packets are sent from the consumer node according to the sending rate described above. The consumer-driven approach has advantages due to its easy deployment and the scalability for growth of network traffic. Because of this, to forward the Interest/Data packets while avoiding congestion, any routers do not need to collect the statistics of incoming packet and control outgoing packet.

Adaptive rate control

As a method for achieving and maintaining low latency transmission and available best throughput on the consumer node, our mechanism controls the sending rate of the Interest packets in order to control throughput of Data packets according to available network bandwidth. This means that the interval time of each Interest packet is adaptively adjusted to the appropriate value by estimating available bandwidth using two values. The first one is the variation of RTT, and the second one is the loss condition of each packet. In order to avoid increasing the queuing delay in the router, our mechanism quickly decreases the sending rate of the Interest packets based on the increase of RTT or packet loss. In the other case, our mechanism conversely increases the sending rate of Interest packets to keep good throughput.

Measuring RTT in each short period

To contend with source variation, our mechanism presumes whether the network is congested or not, by comparing the average RTT of the Data packets which are received in a constant period. As it relates to the variation of the location of the source, we believe the RTT variation is conspicuously drastic compared to network congestion. To presumes whether the congestion or the source variation, in our algorithm, we do not use a weighted average RTT. A weighted average RTT is sometimes useful for elimination of measuring error and smooth rate adaptation. However, it is influenced by the variation of the source for a long term. On the other hand, the average measured RTT in each period that enables us to identify the variation of the source or network congestion. To distinguish these reasons for the RTT variation definitely, the statistical time unit (to calculate average RTT) must be set to a short period.

Maintaining RTT Fairness

In order to satisfy RTT Fairness, our rate control mechanism periodically estimates available bandwidth based on RTT variation and the packet loss condition. It is possible to obtain the same characteristic of rate control, regardless of differences in delay of network path. In the case of ACK (Acknowledgment) based rate control mechanism, such as TCP or AIMD, they are tremendously affected by the differences in delay of network path, because the rate control timing for each consumer node is synchronized to the timing of the packet reception. As a result, it is difficult to satisfy RTT Fairness. Therefore, we adopt a scheme which estimates available bandwidth and controls the sending rate of the Interest packets in each constant period.

3.4. Algorithm Description

In this algorithm, for every constant period (P), pps_t which means the number of Interest packets sent per second at the time (t), is calculated. A consumer node sends Interest packets according to the sending interval (I_t) with the formula (1,2). k is the correction factor of the Data packet size; s [byte] is the average Data packet size; χ is the pre-defined constant value. In addition, pps_t multiplied by $8s$ makes a value of the bandwidth estimation [bps]. In formula (2), s may be greatly different among applications, and then bandwidth consumption of their Data packets are different when each application sends Interest packets with same pps_t. k is used to eliminate difference in packet size among competing Data packet flows. I_t is increase with growth s compared with other flows. As a result, the applications, which use larger Data packet size, decrease Interest sending rate.

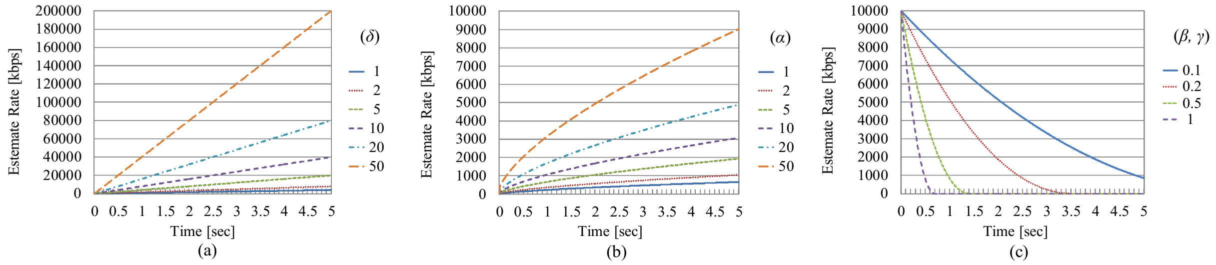

Fig. 2 Numerical simulation of algorithm parameter (P=10ms, s=1024byte)

$$I_t = \frac{1}{k \cdot pps_t} \quad (1)$$

$$k = \frac{\chi}{s} \quad (2)$$

There are two phases in the rate control algorithm. Algotithm1 is common preprocess for each phase that measures RTT and identifies network conditions for each P. "*Start phase*" in Algorithm2 is corresponding to the phase when the communication begins, and "*Adaptive phase*" in Algorithm3 is corresponding to the phase after the "*Start phase*". The objective of "*Start phase*" is the adjustment of throughput to available bandwidth. When a video streaming application starts the communication, there is no way to estimate the available bandwidth precisely. At the same time, the application should quickly obtain suitable video quality. In "*Start phase*" – compared to "*Adaptive phase*" – the algorithm tries to roughly and quickly adjust the initial pps_t to the available bandwidth. If the algorithm detects the symptom of the congestion, it promptly moves from "*Start phase*" to "*Adaptive phase*". The objective of "*Adaptive phase*" is to adapt throughput to the variation of available bandwidth while maintaining RTT Fairness during the session. In this phase, the important point is the algorithm of controlling pps_t, which makes it possible to maintain RTT Fairness. Additionally, as already described in section 3.2, the network throughput which is provided with controlling pps_t directly affects the video quality and video stream bit-rate. From the viewpoint of video quality, the rapid change of pps_t is undesirable. In order to realize these requirements, we defined the algorithm that controls pps_t smoothly with the square root – based on the principle which is described in [25].

Next, we describe the parameters of controlling pps_t in Algorithms2 and 3. Fig. 2(a) shows the increase of the bandwidth estimation according to δ in "*Start phase*". "*Start phase*" increases the estimation in a linear manner, and the adjustment of the estimation to available bandwidth is shorter with increasing δ. However, the larger δ may cause serious congestion when the phase changes. Fig. 2(b) shows the increase of the bandwidth estimation according to α in "*Adaptive phase*". In "*Adaptive phase*", the larger α improve adaptability to increase of available bandwidth. At the same time, the larger α is easy to increase queuing delay and packet loss in a bottleneck link. Fig. 2(c) also shows the reduction of the bandwidth estimation according to β and γ in "*Adaptive phase*". The larger β and γ, which are employed for reduction of pps_t when detect the symptom of the congestion, improve adaptability to congested network situation, and make possible to avoid queuing delay increases and packet loss. Meanwhile, the larger β and γ tend to cause the unnecessary reduction in the estimation. Furthermore, pps_t directly affects the video stream bit-rate, and then the unnecessary change of pps_t causes the undesirable decrease of video quality. On designing these parameters, it is necessary to consider above trade-off and affects for video quality.

Table 1 Algorithm notation

Ip_i	sequence number: i of Interest packet
Dp_i	Data packet for Ip_i
RTT_i	RTT between (Ip_i, Dp_i)
$latest\_seq_t$	latest sequence number of received Data packet in time: t
$jitter\_offset$	offset of normal network delay vibration

Algorithm1: Rate control in common preprocess

for i in $latest\_seq_{t-P}$ to $lastest\_seq_t$
  if Dp_i is received then
    $RTT\min \leftarrow \min(RTT\min, RTT_i)$
    $recv\_ct_t \leftarrow recv\_ct_t + 1$
    $SumRTT_t \leftarrow SumRTT_t + RTT_i$
  else
    $unrecv\_ct_t \leftarrow unrecv\_ct_t + 1$
  end if

end for

$AvgRTT_t \leftarrow \dfrac{SumRTT_t}{recv\_ct_t}$

if $AvgRTT_t \le (RTT\min + jitter\_offset)$ then

  $case1 \leftarrow true$

else if $AvgRTT_t \le AvgRTT_{t-P}$ then

  $case2 \leftarrow true$

else

  $case3 \leftarrow true$

end if

Algorithm2: Rate control in "*Start phase*"

if *case1* or $unrecv\_ct_t = 0$ then

  $pps_t \leftarrow pps_{t-P} + \delta$

else

  $pps_t \leftarrow pps_{t-P} / 2$

  Go to "Adaptive phase"

end if

Algorithm3: Rate control in "*Adaptive phase*"

if $unrecv\_ct_t = 0$ then

  if *case1* or *case2* then

    if not *congestion_sign* then

      $pps_t \leftarrow pps_{t-P} + \alpha / \sqrt{pps_{t-P}}$

    else

      $congestion\_sign \leftarrow false$

    end if

  else if *case3* then

    if *congestion_sign* then

      $pps_t \leftarrow pps_{t-P} - \beta \cdot \sqrt{pps_{t-P}}$

    else

      $congestion\_sign \leftarrow true$

    end if

  end if

else

  $pps_t \leftarrow pps_{t-P} - \gamma \cdot \sqrt{pps_{t-P}}$

end if

4. Simulation results

In this section, we present the simulation results of the proposed mechanism with ndnSIM [26]. ndnSIM is the NDN simulation module based on ns-3 [27] and released by the NDN project [1]. In this evaluation, there are a few assumptions that clarify the basic characteristic of the rate control algorithm:

- Each consumer node requests content with sequential numbering in the Content Name for each Interest packet.
- Each consumer node has determined the Content Name to fetch by other means.
- Each publisher node provides single video stream with variable bit-rate.

Unless otherwise noted, we use the parameters of the algorithm in Table2. In this evaluation, on selecting α, β, γ and δ, we assumed the range of video stream bit-rate is from several hundreds kbps to 10Mbps, such as SD (Standard Definition) and HD (High Definition) video quality. P is selected in consideration of timer interrupt precision that general Operating System (OS) can provide to an application program. χ and s are set to the same value to simplify the evaluation. We also assume that any bottleneck links has enough link speed for transmitting minimum bit-rate of video stream.

Table 2 Algorithm parameter

P	10ms
jitter_offset	10ms
α	20
β	0.5
γ	0.5
δ	20
χ	1024
s	1024

4.1. Latency & bandwidth efficiency

First we evaluate the basic function of the bandwidth estimation while keeping low-latency transmission and good throughput using the single bottleneck link in Fig. 3 and Table 3. Each consumer node accesses the different publisher node, which has distinct named content, transmitting an Interest packet for it.

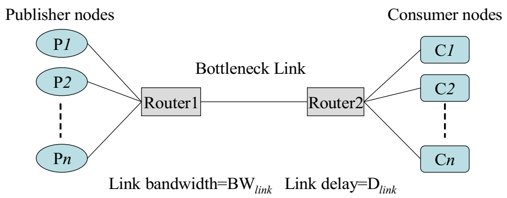

Fig. 3 Single bottleneck link topology

Table 3 Network topology parameter

$BW_{Pn\text{-}R1}$	1Gbps
$D_{Pn\text{-}R1}$	1ms
$BW_{R2\text{-}Cn}$	1Gbps
$D_{R2\text{-}Cn}$	1ms
Queue	Droptail
Queue Size	50pkt

Fig. 4 shows the result of bandwidth estimation and average RTT in each P, with n=1, $BW_{R1\text{-}R2}$=10Mbps, $D_{R1\text{-}R2}$=8ms. From Fig. 4, we can see that our proposed mechanism effectively estimates its available bandwidth while keeping low latency transmission.

Next, we evaluate (as above) if $D_{R1\text{-}R2}$ is varied from 3ms to 148ms. Figs. 5 and 6 summarize the data about rate of bandwidth utilization and average increase of transmission delay (RTT). To compare performance with our methods, we also show the evaluation result of consumer-driven approach mechanism that controls Interest packet rate with AIMD in the same manner. From Figs. 5 and 6, we can see that transmission with our mechanism shows better throughput and lower latency regardless of the network delay. Meanwhile, in the case of AIMD, it cannot satisfy both of the above performances at the same time.

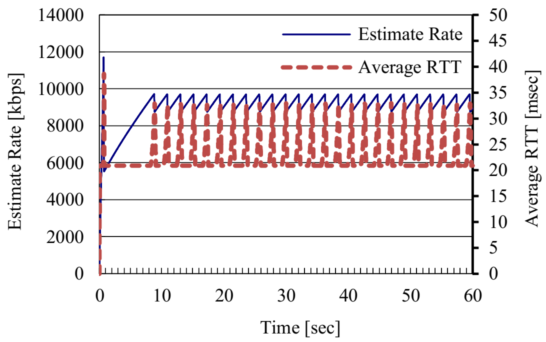

Fig. 4 Bandwidth estimation and average RTT (n=1, $BW_{R1\text{-}R2}$=10Mbps, $D_{R1\text{-}R2}$=8ms)

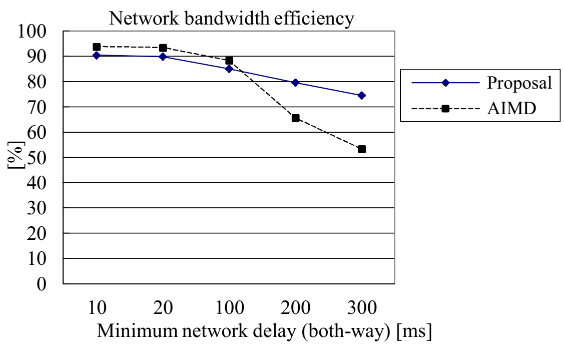

Fig. 5 Network bandwidth efficiency (n=1, $BW_{R1\text{-}R2}$=10Mbps)

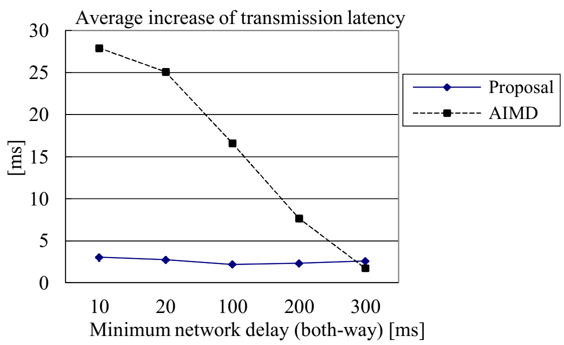

Fig. 6 Average increase of transmission latency (n=1, $BW_{R1\text{-}R2}$=10Mbps)

4.2. Adaptability

We evaluated the adaptability of our rate control algorithm to bandwidth variation on the topology shown in Fig. 3. In order to verify the adaptability, the characteristic of the bandwidth estimation and the variation of transmission latency are evaluated with changing the bandwidth of the bottleneck link ($BW_{R1\text{-}R2}$) simply at a point of time.

Fig. 7 is the results of bandwidth estimation and average RTT in each P when $BW_{R1\text{-}R2}$ is changed from 10Mbps to 5Mbps. This figure shows that our mechanism rapidly decreases the estimated rate in response to the sudden increase of the average RTT when network bandwidth is decreased. As a result, the average RTT becomes the stable condition. In this way, our mechanism adapts the rate to decreasing network bandwidth if it varies drastically. Next, we evaluate the adaptability if $D_{R1\text{-}R2}$ is varied from 3ms to 148ms. Fig. 8 summarizes the convergence time which indicates that the estimated rate is under the decreased network bandwidth. This result is average in five times simulation that was varied the time to decrease network bandwidth. From Fig. 8, in the case of our proposal, the trend of the convergence time is almost the same. In the case of AIMD, it dynamically decreases the window size, when it detects congestion. As a result, the convergence time is shorter than in our proposal.

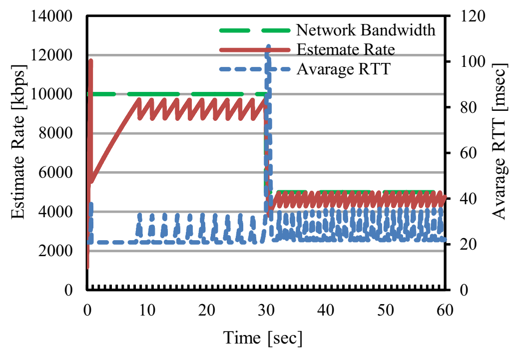

Fig. 7 Bandwidth estimation and average RTT (n=1, $BW_{R1\text{-}R2}$= from 10Mbps to 5Mbps, $D_{R1\text{-}R2}$=8ms)

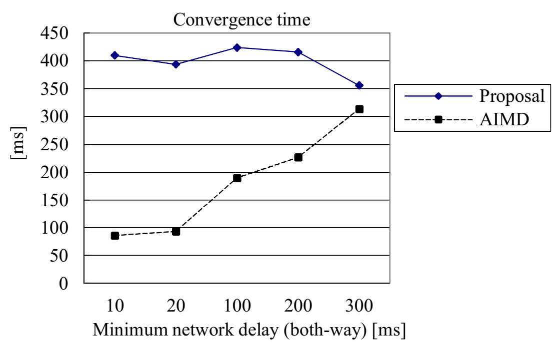

Fig. 8 Convergence time (n=1, $BW_{R1\text{-}R2}$= from 10Mbps to 5Mbps)

Fig. 9 is the bandwidth estimation and average RTT in each *P* when $BW_{R1\text{-}R2}$ is changed from 5Mbps to 10Mbps. From Fig. 9, we can see that our mechanism increases the estimated rate gradually without increasing average RTT, and then our mechanism adapts it to increased network bandwidth. Next, we evaluate (as above) if $D_{R1\text{-}R2}$ is varied from 3ms to 148ms. Fig. 10 summarizes the convergence time, which indicates that the estimated rate increases to near the increased network bandwidth. This result is average in five times simulation that was varied the time to increase network bandwidth. From Fig. 10, in the case of our proposal, the trend of the convergence time is almost the same. In the case of AIMD, the convergence time is shorter than in our proposal when the network delay is short-range, because AIMD increases the rate aggressively than in our proposal. However, the convergence time becomes longer than in our proposal when the network delay is long-range (e.g., more than 100ms).

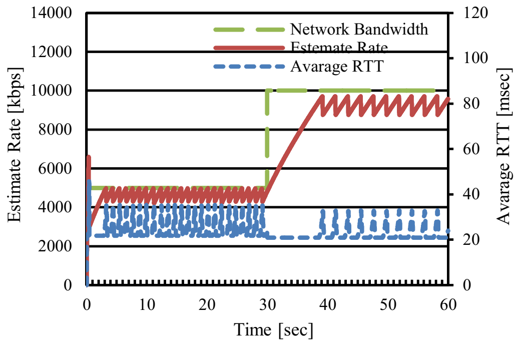

Fig. 9 Bandwidth estimation and average RTT (n=1, $BW_{R1\text{-}R2}$= from 5Mbps to 10Mbps, $D_{R1\text{-}R2}$=8ms)

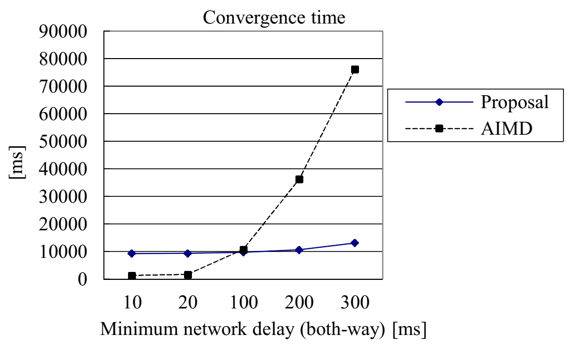

Fig. 10 Convergence time (n=1, $BW_{R1\text{-}R2}$= from 5Mbps to 10Mbps)

4.3. RTT Fairness

We evaluated the RTT Fairness on the topology shown in Fig. 3. Fig. 11 is the results of estimated bandwidth on each consumer node, with n=4, $BW_{R1\text{-}R2}$=10Mbps, $D_{R1\text{-}R2}$= 8ms. This result shows that the algorithm effectively and fairly estimates its available bandwidth. In this way, when each publisher-consumer node pair has the same link delay in Fig. 3, each consumer node can get throughput fairly.

The next evaluation compares the throughput of consumer nodes – whether the link delay between the Router2 and each consumer node is the same or not, as in Fig. 3. Fig. 12 is the histogram of average throughput for each consumer node in 60 seconds, when n=32, $BW_{R1\text{-}R2}$ =100Mbps, $D_{R1\text{-}R2}$=8ms and all $D_{R2\text{-}Cn}$=1ms. Fig. 13 shows average throughput where each $D_{R2\text{-}Cn}$ varies based on the number of nodes, i.e., 5(n-1)+1[ms]. To compare the performance, we also show an evaluation results of AIMD in the same manner. From Fig. 13, in the case of AIMD, throughput of each consumer node is widely dispersed when each $D_{R2\text{-}Cn}$ is different, which indicates that a mechanism based on AIMD cannot provide RTT Fairness. On the other hand, these results show that our proposed mechanism sufficiently provides RTT Fairness.

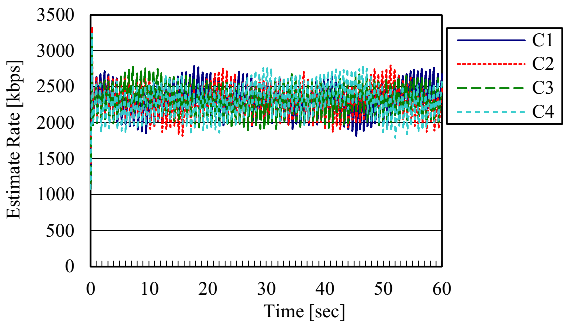

Fig. 11 Bandwidth estimation (n=4, $BW_{R1\text{-}R2}$=10Mbps, $D_{R1\text{-}R2}$=8ms)

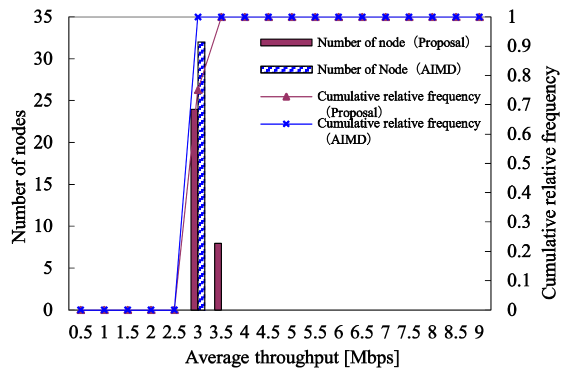

Fig. 12 Histogram of average throughput (n=32, $BW_{R1\text{-}R2}$=100Mbps, $D_{R1\text{-}R2}$=8ms, $D_{R2\text{-}Cn}$=1ms)

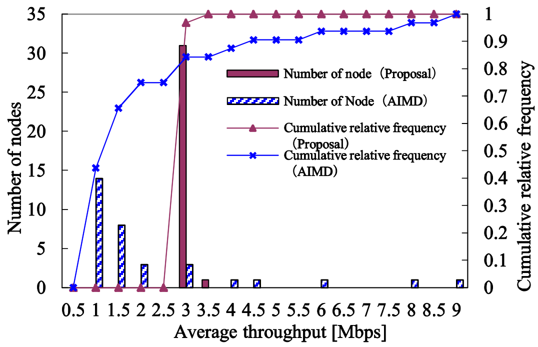

Fig. 13 Histogram of average throughput (n=32, $BW_{R1\text{-}R2}$=100Mbps, $D_{R1\text{-}R2}$=8ms, $D_{R2\text{-}Cn}$=5(n-1)+1[ms])

4.4. Effectiveness for source variation

We evaluated the effectiveness for the source variation on the topology shown in Fig. 1. In this simulation, we assume that each consumer node (C1, C2 and C3) simultaneously start to send Interest packets for same Data packet which is generated by a publisher node as shown in Fig. 1(a). In order to generate the source variation simply, C2 stops sending Interest packets while receiving Data packet as shown in Fig. 1(c). After C2 stop sending Interest packet, the source of Data packets for C3 shifts from Router2 to Router1, and then the RTT which is measured on C3 becomes long than in Fig. 1(a). Fig. 14 shows the throughput of both our mechanism and AIMD on C3, when all links are same bandwidth (=10Mbps) and same delay (=10ms). C2 stops sending Interest packet at 30 seconds. This results shows our mechanism can keep good throughput regardless of the source variation. This is because our proposal identifies the source variation, and estimates available bandwidth correctly. On the other hand, the throughput of AIMD significantly decreases after source variation. This is because the throughput of AIMD decreases according to increase of network delay.

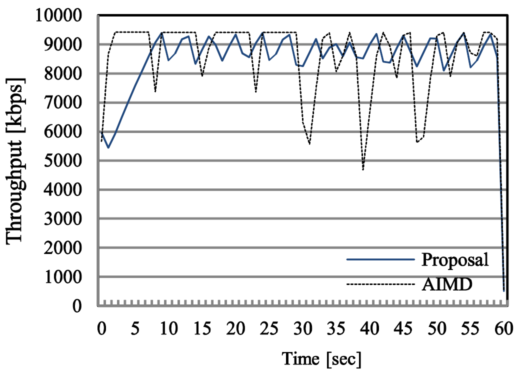

Fig. 14 Results of throughput

4.5. Consideration

In this section, we evaluated proposal transport mechanism and AIMD as comparative method. For real-time video streaming, the important point is keeping not only low latency transmission but also good throughput which directly affects the video quality. Our mechanism keeps better throughput and lower latency than in AIMD, regardless of the network delay. We also show our mechanism identifies the source variation, and estimates available bandwidth correctly. Furthermore, our mechanism fairly shares the network bandwidth among several competing consumer nodes, regardless of delay of their network path. In the same situation, AIMD cannot share the network bandwidth fairly, because throughput of AIMD significantly affected by the delay of their network path.

On the other hand, AIMD can adapt to network bandwidth change immediately than in our mechanism. As described in section 3.4, our mechanism can improve the adaptability by modifying the algorithm parameters. However, we think excessive adaptability is undesirable for video streaming applications [25].

5. Conclusion and Future work

For real-time video streaming in NDN, this paper focused on low-latency transmission, maintaining available best throughput, and considered unexpected RTT valuation, RTT Fairness (which indicates sharing network bandwidth fairly in a bottleneck link regardless of network delay between publisher node and consumer node). In order to realize these objectives, we proposed consumer-driven adaptive rate control with a bandwidth estimation method that uses average RTT in short periods instead of smoothed RTT. This mechanism has advantages for its easy deployment and scalability for growth of network traffic, because it does not require any router supports. To show effects for real-time video streaming, we evaluated it and AIMD with the network simulator. The simulation results showed that our mechanism works well for estimating available bandwidth while maintaining good throughput, low-latency, and RTT Fairness, which is important in NDN because RTT would be easily changed based on the cached content availability on its path. On the other hand, AIMD cannot maintain good throughput and low-latency at the same time, regardless of the network delay. In addition, it cannot provide RTT Fairness. This is because that

AIMD algorithm cannot obtain good throughput in long network delay environment essentially.

For our future work, we plan to explore the adaption for multi-path and multi-interface situations in a heterogeneous network.

Acknowledgments
We would like to express our gratitude to Professor Lixia Zhang and team members of REMAP, University California, Los Angeles.

References

[1] NDN project. http://www.named-data.net/.

[2] Van Jacobson, Diana K. Smetters, James D. Thornton, Michael F. Plassi, Nicholas H. Briggs, and Rebecca L. Braynard, “Networking Named Content,” ACM CoNEXT, 2009.

[3] Henderson T.R., E. Sahouria, and R.H. Katz, “On Improving the Fairness of TCP Congestion Avoidance,” IEEE GLOBECOM, 1998.

[4] Sally Floyd, “Connections with Multiple Congested Gateways in Packet-Switchd Networks Part 1: One-way Traffic,” Computer Communication Review, Vol.21, No.5, October 1991, pp.30-47.

[5] Cisco Systems, Inc, Cisco Visual Networking Index: Forecast and Methodology, 2012-2017, 2013.

[6] Jake Dunagan and Mike Liebhold, “The Future of Real-Time Video Communication,” http://www.iftf.org/uploads/media/SR1278_Real-TimeVideoCommunication_2.12.sm.pdf, 2009.

[7] Giovanna Carofiglio, Massimo Gallo, and Luca Muscariello, “Icp: Design and evaluation of an interest control protocol for content-centric networking,” INFOCOM NOMEN Workshop, 2012.

[8] Stefano Salsano, Andrea Detti, Matteo Cancellier, Matteo Pomposini, and Nicola Blefari-Melazzi, “Transport-layer issues in information centric networks,” ACM SIGCOMM ICN Workshop, 2012.

[9] Somaya Arianfar, Pekka Nikander, Lars Eggert, Jorg Ott, and Walter Wong, “Contug: A receiver-driven transport protocol for content centric networks,” IEEE ICNP, 2010.

[10] Dah-Ming Chiu and Raj Jain, “Analysis of increase and decrease algorithms for congestion avoidance in computer networks,” Computer Networks and ISDN systems, 17, 1989.

[11] Giovanna Carofiglio, Massimo Gallo, and Luca Muscariello, “Joint hop by hop and receiver-driven interest control protocol for content-centric networks,” ACM SIGCOMM ICN Workshop, 2012.

[12] Giovanna Carofiglio, Vinicius Gehlen, and Diego Perino, “Experimental Evaluation of Memory Management in Content-Centric Networking,” International Conference on Communications, 2011.

[13] M. Tortelli, I. Cianci, L. A. Grieco, G. Boggia, and P. Camarda, “A Fairness Analysis of Content Centric Networks,” International Conference on the Network of the Future, 2011.

[14] Giovanna Carofiglio, Massimo Gallo, and Luca Muscariello, “Bandwidth and Storage Sharing Performance in Information Centric Networking,” ACM ICN, 2011.

[15] Ciancaglini V., Piro G., Loti R. Grieco L.A. and Liquori L, “CCN-TV: A Data-centric Approach to Real-Time Video Services,” Advanced Information Networking and Applications Workshops, 2013.

[16] Derek Kulinski, and Jeff Burke, “NDNVideo: Random-access Live and Pre-recorded Streaming using NDN,” In Technical Report, http://named-data.net/techreport/TR007-streaming.pdf, 2012.

[17] Reinhard Grandl, Kai Su and Cedric Westphal, “On the Interaction of Adaptive Video Streaming with Content-Centric Networking,” Packet Video Workshop, 2013.

[18] Stefan Lederer, Christopher Muller, Benjamin Rainer, Christian Timmerer and Hermann Hellwagner, “An Experimental Analysis of Dynamic Adaptive Streaming over HTTP in Content Centric Networks,” In Proceeding of the IEEE International Conference on Multimedia and Expo, 2013.

[19] Stefan Lederer, Christopher Muller, Benjamin Rainer, Christian Timmerer and Hermann Hellwagner, “Adaptive Streaming over Content Centric Networks in Mobile Networks using Multiple Links,” In Proceeding of the IEEE International Workshop on Immersive & Interactive Multimedia Communication over the Future Internet, 2013.

[20] MPEG-DASH, http://dashif.org/.

[21] NDN Packet Format Specification: Interest Packet, http://named-data.net/doc/ndn-tlv/interest.html

[22] Van Jacobson, “Congestion avoidance and control,” Proc, ACM SIGCOMM, 1988.

[23] John Nagle, “Congestion Control in IP/TCP Internetworks,” RFC896, 1984.

[24] Floyd.S, “Congestion Control Principles,” RFC2914(BCP:41), 2000.

[25] Deepak Bansal, and Hari Balakrishnan, “Binomial Congestion Control Algorithms,” IEEE INFOCOM, 2001.

[26] ndnSIM, http://www.ndnsim.net/.

[27] ns-3, http://www.nsnam.org/.