Network-Assisted Congestion Control for Information Centric Networking

Anselme Ndikumana, Saeed Ullah, Rossi Kamal, Kyi Thar, Hyo Sung Kang, Seung Il Moon, and Choong Seon Hong*
Department of Computer Engineering, Kyung Hee University
{anselme, saeed, rossi, kyithar, kanghs, moons85, cshong}@khu.ac.kr

***Abstract*—Internet has grown very rapidly in the last couple of decades and still growing because of the expansion and utilization of various services and applications. Consequently, demand of delay and throughput sensitive services, like audio/video is also increasing. Information Centric Networking (ICN) is proposed as an architecture for the future Internet to meet the modern users and application requirements. In ICN users send requests (Interest Packets) for the Data they need. Interest packet is assigned a lifetime, which greatly affects the Quality of Experience (QoE) because user needs to resend the Interest, when the lifetime expires. Interest lifetime may be expired because of congestion, or Interest lifetime is shorter than the network delay, etc. Waiting for the expiration of an Interest lifetime to resend it is merely appropriate for best effort traffic, rather than services which require high throughput and are delay sensitive. In this paper, we propose Network-Assisted Congestion Control mechanism in ICN, which detects the congestion before it happens, and provides notification to downstream node. On reception of the notification, downstream node continuously reduces the traffic rate. However, when the downstream node fails to adjust the sending rate, the same procedure continues, until the sending node reduces the traffic rate through adjusting its congestion window. We have intensively evaluated our proposal by comparing it with similar proposal using ndnSIM. The experimental results show that our proposal achieves up to 59 percent performance improvement over other proposal in the literature.**

Keywords—Information Centric Networking, Network-Assisted Congestion Control, Reduce Sending Rate, Modified Dynamic Token Bucket.

I. Introduction

Huge penetration of the Internet in various fields of life has caused the creation of many applications that generate huge amounts of data. According to CISCO Visual Networking Index (VNI), the Internet video traffic will be 75% in 2018 of all Internet traffic, from 57% in 2013 [1].

To meet the increasing demand for high throughput and delay sensitive applications a new Internet architecture, i.e., Information Centric Networking [2] has been introduced. Content Centric Networking (CCN) [3] is one of the most promising architecture among many ICN proposals. CCN focuses on contents, where contents are requested by names rather than the destination IP addresses of the hosts that have the contents. In CCN, there are two types of the packets, Interest packet, which is used for requesting the content, and data packet, which is transmitted in response to the Interest packet in its reverse path [3]. Each Interest is assigned a lifetime and when the lifetime expires, the Interest gets rejected from Pending Interest Table (PIT) and the consumer needs to resend the Interest [4]. However, waiting for a period of Interest lifetime expiration, is not effective for services that require high throughput and minimum delay. There can be many causes of Interest lifetime expiration like network congestion, unstable output face(s), shorter Interest lifetime than the network delay, etc. Network congestion affects the delay sensitive applications and causes packets loss and retransmission. In order to provide delay sensitive applications better Quality of Service (QoS), early network congestion detection and prevention is very important.

In order to make CCN complete, traffic control is required to ensure that the packets meet negligible delay when they share links [5]. We cannot talk about traffic control and left behind buffer, instead of dropping the packets, the buffer accommodates temporally transient traffic for making sure that the link does not become idle [6].

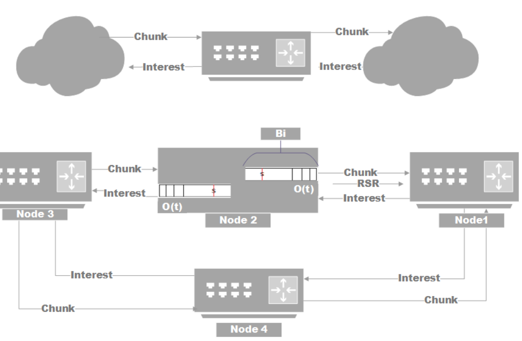

Fig. 1. Network-Assisted Congestion Control mechanism

In CCN, congestion might occur in the network as chunks overflow the transmission buffer [7]. In other words, packets arrive faster than the router's processing capacity or more than the out-link capacity, which causes the buffer fullness. In this paper, we have proposed Network-Assisted Congestion Control mechanism (NACC), which detects the congestion before it happens, and provides an indication to downstream

This research was supported by Basic Science Research Program through National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2014R1A2A2A01005900). *Dr. CS Hong is the corresponding author.

node through Reduce Sending Rate (RSR) message. On reception of RSR message, downstream node uses Modified Dynamic Token Bucket (MDTB) to reduce the traffic rate until it does not continue to receive RSR. When the downstream node fails to adjust the sending rate, the same procedure continues until RSR reaches the sending node. When sending node receives the RSR message, it reduces the traffic rate by adjusting its congestion window. Our simulation results show that our proposal prevents packets drop due to buffer overflow, and performs well with slow start and Additive Increase/ Multiplicative Decrease (AIMD).

The rest of this paper is organized as follows, Section II describes related work, while Section III describes Network Assisted Congestion Control mechanism. Section IV provides evaluation, which includes experimental setup and results. We conclude the paper with some future directions in section V.

II. Related work

In this section, we discuss on some important related works. In [8], when the Interest is sent, the receiver sets a timer. If the timer expires without getting the Data, it assumes congestion has occurred and hence, decreases the receiver window. In additional to that, at every outgoing face, the Interest rate control is carried-out with the help of credit counter, when the flow is not bottlenecked the Interest is forwarded to upstream node, otherwise the Interest is queued in drop tail following FIFO (First-In, First-Out) queuing model. In [5], the authors proposed per-flow queuing and overload control, with the help of Deficit Round Robin (DRR), the router dropped the packets from the longest queue in case of buffer overflow. However, when a packet is rejected, the packet payload is discarded and its header is modified and returned to the receiver as a signal of congestion. This helps the consumer node to detect the loss and retransmit the lost packet. In [9], on the reception of each Interest, router verifies weather it has the chunk(s) intended to be requested after, and appends this information on returning chunk. Based on that, the consumer node keeps multiple timeout values for each flow in order to predict the location of the chunks before being requested and estimate accurately the retransmission timeouts. In [7], each node monitors the level of chunks in the router's transmission buffer, computes the associated Interests, and maintains the transmission queue around the fixed threshold. In additional to that, the authors introduced the shaping delay in which the Interests have to meet, but in case of buffer overflow, the Interest shaping allows to drop the packets anticipatively.

III. Network-Assisted Congestion Control mechanism

In this section, we discuss on MDTB as a queue scheduling algorithm for active flows, fair resource allocation and traffic shaping. On the receiver side, we discuss AIMD; which is based on RSR Message, in additional to chunk received and Interest timeout. In our Network-Assisted Congestion control proposal presented in the Figure 1, we have one consumer (node 1) and one content provider (node 3). Content is not cached in node 2. Node 2 forwards all interests to node 3. When node 2 reaches to its buffer fullness threshold, and it is also not able to allocate incoming packets to network face buffer; it notifies its downstream (node 1) through RSR message. On reception of RSR message, node 1 reduces traffic rate in which it sends packets to node 2, and adjust (increase) the traffic rate of face connecting node 1 to node 4.

A. *Modified Dynamic Token Bucket (MDTB) Algorithm*

Network traffic is generated differently in terms of sizes, behaviors and priorities. Due to the simplicity of a DTB [11] in terms of computation and fair bandwidth allocation, we propose a new algorithm named MDTB, as a modified version of DTB for ICN. MDTB controls the rate at which packets are injected into the network, in order to avoid unnecessary packet delay, retransmission and loss. It provides a fair share of buffer space, which depends on buffer capacity and the number of active flows. . In order to simplify memory management, in this paper, we consider that the entire buffer is partitioned into a number of buffers having a fixed size, where each buffer with respective size is associated to each face. In ICN, a flow becomes active when it starts sending out its first Interest packet, otherwise it becomes inactive. One flow may have many transient packets, and if the size of the flow is equal to the buffer threshold S, the node checks the utilization of other buffer(s) associated to other face(s) and allocates new incoming packets accordingly, in order to make sure that all the buffers associated to faces are well utilized, and none is overloaded while others are underutilization.

In this paper, we consider M/M/1/B finite buffer queuing system [12], where the packet arrives according to Poisson process with arrival rate λ. Inter-arrivals are Independent and Identically Distributed (IID) and are exponentially distributed. The departure process is also a Poisson process with exponential distribution and service rate μ. In order to avoid buffer overflow, we use buffer threshold S which is less than 50% of the total face buffer. When the threshold is reached and the node is unable to allocate the packets to another buffer, it notifies its downstream node through RSR message. The probability of n packets in system

$$P_n = (1-\rho)(\rho)^n/(1-\rho^{B+1}) = \rho^n P_o \quad (1)$$

Where

$$P_0 = (1-\rho)/(1-\rho^{B+1}) \quad (2)$$

The loss probability P_b (an incoming packet sees B packets in the system) is given in the following equation.

$$P_b = (1-\rho)(\rho)^B/(1-\rho^{B+1}) \quad (3)$$

For the queue occupancy, we use $O(t)$ as an expected number of interests and chunks on transmission queue.

$$O(t) = (\rho/(1-\rho))(((B+1)/(1-\rho^{B+1}))(\rho^{B+1})) \quad (4)$$

For larger buffer size, node keeps loss probability less than 1%, and from P_b the node computes the required B for each flow. For equal share of bottleneck link, each node will assign a fair rate $R(t)$ to all outgoing interests as well as returning chunks passing through it. The rate depends on link capacity

C, number of flows $N(t)$ and buffer size associated to the face B_i, for i = 1, 2, ..., n.

$$R(t) = (C + (B_i - O(t)))/N(t) \quad (5)$$

On the arrival of packet to the buffer, the packet l_i is transmitted if there are enough tokens to cover its length.

- For the incoming packet i, the node allocates buffer B to packet until $l_i \leq R(t) \times \Delta t$, where l_i is a Length of the packet i.
- If the difference between buffer threshold S associated to face B_i and queue occupancy $O(t)$ is zero ($S - O(t) = 0$), then the nodes allocates the incoming packet to another face buffer $B_j \ for \ B_j \neq B_i$, based on its queue occupancy.
- Else if node fails to allocate the incoming packets to another face buffer, it returns RSR message to downstream node

In this paper, for congestion avoidance, we opt the optimal face buffer B_i for L-model as as proposed in [13]. The buffer (B) allocated to the packets must be less than or equal to B_i

$$B_i = 2C * RTT \quad (6)$$

B. AIMD and Reduce Sending Rate (RSR) Message

In this subsection, we introduce congestion control in ICN with slow start and AIMD, as a modified version of existing TCP congestion control [15]. In ICN, due to caching functionality, where the source of the content keeps changing its location, it becomes harder to calculate the exact Round-Trip Time (RTT). In ICN, RTT is defined as a length of time for sending Interest packet and receiving correspondent chunk. Returning chunk acts as the acknowledgment of a received Interest, and based on which consumer node keeps updating RTT and timeout values. We consider two important parameters; congestion window (cwnd) and slow start threshold (ssthresh), in which depend on a received chunk, RSR message, and Interest timeout. The modified version of AIMD for ICN is summarized and explained more in the details in the following 5 steps:

- *Step 1:* During the slow start (cwnd $\leq$ ssthresh), cwnd starts from one Interest per RTT, and ssthresh sets to infinity. For each chunk received, the congestion window is incremented by one Interest.
- *Step 2:* On reception of RSR message, one half of the current cwnd will be saved as slow start threshold (ssthresh).
- *Step 3:* When Interest does not return data within the timeout period, one third of the current cwnd will be saved as slow start threshold (ssthresh), then the retransmitted failed Interest.
- *Step 4:* When cwnd $\geq$ sshreshold, the node starts congestion avoidance, then the cwnd will be incremented by one Interest per RTT.
- Step 5: On current slow start threshold (ssthresh), when a retransmitted Interest does not return data, the current cwnd restarts from the step 1. Retransmit failed Interest and start double timeout value

IV. Evaluation

A. Experimental Setup

In this section, we analyze the performance of our NACC proposal for ICN through simulations. All our simulations were performed using the ndnSIM 2.0 [14], an ns-3 based NDN simulator. Figure 2 shows our simulation scenario, where

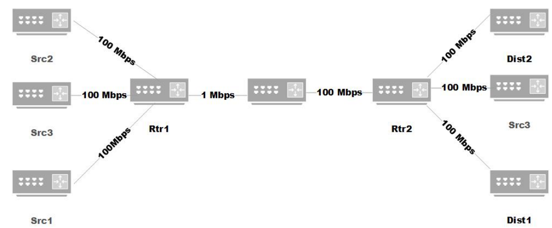

Fig. 2. Simulation Scenario

we have three consumers, Src1 requests content with prefix /Flow1, Src2 requests content with prefix /Flow2 and Src3 requests content with prefix /Flow3. On the other side, we have three content providers, Dist1 replies all requests for prefix /Flow1, Dist2 replies all requests for prefix /Flow2 and Dist3 replies all requests for prefix /Flow3. All links have $10ms$ propagation delay and $100Mbps$ bandwidth, except the link between router Rtr1, Rtr2, which is $1Mbps$. Payload size is 1000 bytes. For buffer size, we consider buffer associated to each face equals to 100 packets. 50 packets were considered as buffer threshold. The content store was set to 100 chunks. In this scenario, we evaluated our proposal NACC by comparing it with existing similar proposal "Hop by Hop Interest Shaping mechanism for CCN (HoBHIS)"[7]. BestRoute was used as forwarding strategy, and on consumer side, two existing interest traffic generation applications, namely ConsumerCbr and ConsumerWindow [10] were used.

B. Experimental Results

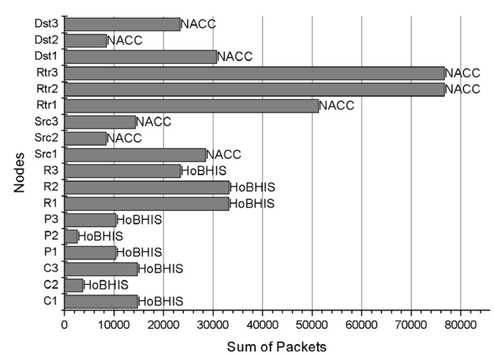

Fig. 3. Number of packets sent per node and per mechanism with ConsumerCbr

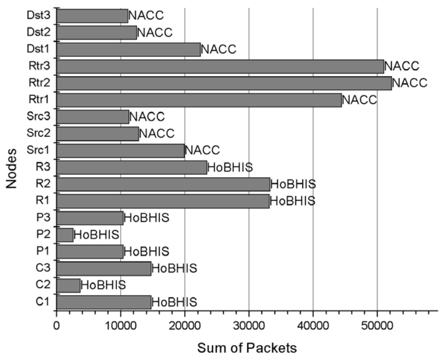

Fig. 4. Number of packets sent per node and per mechanism with ConsumerWindow and ConsumerCbr

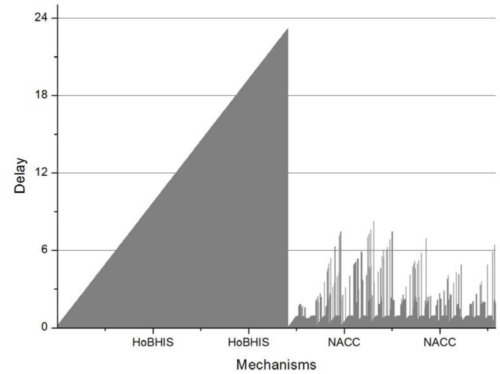

Fig. 5. Delay (seconds) experienced by packets per mechanism with ConsumerCbr and ConsumerWindow

Figures 3 shows a comparison between HoBHIS and NACC, where a consumer application for both mechanisms were ConsumerCbr. In both mechanisms, Interests were generated with a frequency of 100 Interests per seconds for each consumer, except consumer Src1, which is 25 Interests per seconds. During 50 seconds of the simulation, NACC has sent and received many packets with higher throughput than HoBHIS. On the other side, there was no packet drop for HoBHIS, while NACC experienced packets drop. For NACC, the packet drop is due to the fact that there was no communication between a node and its downstream, consequently there was no possibility of reducing the traffic rate on downstream node.

In Figures 4 and 5, NACC uses ConsumerWindow application with slow start and AIMD; on each chunk received, the congestion window is incremented by one Interest. On reception of RSR message or on Interest Timeout, the sending node (through AIMD) or intermediate node (through MDTB) reduces the transmission rate. On another side, HoBHIS keeps using ConsumerCbr application, with the same setting as described above. The simulation results show that NACC performs better in terms of delay and throughput than HoBHIS, and there were no packets drop in both HoBHIS and NACC.

V. Conclusion and Future directions

In this paper, we presented Network-Assisted Congestion Control mechanism for Information Centric Networking. The proposed scheme prevents congestion before it happens. We introduced Reduce Sending Rate message and Modified Dynamic Token Bucket algorithm for congestion detection and prevention. Our simulation results show that our proposal prevents packets drop due to buffer overflow, and performs well with slow start, AIMD and MDTB. In future, we aim to extend our Network-Assisted Congestion Control mechanism analysis in terms of minimizing packet loss/delay and also to analyze the proposed mechanism in more diversified environment.

References

[1] http://www.cisco.com/web/solutions/sp/vni/vni_forecast_highlights/index.html. Visited on April 14, 2015 at 02:38 PM

[2] Ahlgren, B., Dannewitz, C., Imbrenda, C., Kutscher, D., & Ohlman, B. (2012). "A survey of information-centric networking." Communications Magazine, IEEE, 50(7), 26-36.

[3] Jacobson, V., Smetters, D. K., Thornton, J. D., Plass, M. F., Briggs, N. H., & Braynard, R. L. (2009, December). "Networking named content." In Proceedings of the 5th international conference on Emerging networking experiments and technologies (pp. 1-12). ACM.

[4] Yi, Cheng, et al. "A case for stateful forwarding plane." Computer Communications 36.7 (2013): 779-791.

[5] Oueslati, Sara, James Roberts, and Nada Sbihi. "Flow-aware traffic control for a content-centric network." INFOCOM, 2012 Proceedings IEEE. IEEE, 2012

[6] Wischik, Damon, and Nick McKeown. " Part I: Buffer sizes for core routers." ACM SIGCOMM Computer Communication Review 35.3 (2005): 75-78.

[7] Rozhnova, Natalya, and Serge Fdida. " An effective hop-by-hop Interest shaping mechanism for CCN communications." Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on. IEEE, 2012

[8] Carofiglio, Giovanna, Massimo Gallo, and Luca Muscariello. "Joint hop-by-hop and receiver (consumer)-driven Interest control protocol for content-centric networks." Proceedings of the second edition of the ICN workshop on Information-centric networking. ACM, 2012.

[9] Saino, Lorenzo, Cosmin Cocora, and George Pavlou. "CCTCP: A scalable receiver-driven congestion control protocol for content centric networking."Communications (ICC), 2013 IEEE International Conference on. IEEE, 2013.

[10] http://ndnsim.net/1.0/applications.html. Visited on April 14, 2015 at 10:38 PM

[11] Kidambi, Jayakrishna, Dipak Ghosal, and Biswanath Mukherjee. "Dynamic token bucket (DTB): a fair bandwidth allocation algorithm for high-speed networks."Journal of High Speed Networks 9.2 (2000): 67-87.

[12] Heckmann, Oliver M. "The competitive Internet service provider: network architecture, interconnection, traffic engineering and network design." John Wiley and Sons, 2007.

[13] Prasad, Ravi S., Constantine Dovrolis, and Marina Thottan. "Router buffer sizing for TCP traffic and the role of the output/input capacity ratio." IEEE/ACM Transactions on Networking (TON) 17.5 (2009): 1645-1658.

[14] Afanasyev, Alexander, et al. "NFD developers guide." Technical Report NDN-0021, NDN, 2014.

[15] Allman, Mark, Vern Paxson, and William Stevens. "RFC 2581: TCP congestion control." (1999).