An Improved Interest Forwarding for Content Centric Networking (CCN)

Anselme Ndikumana, Da Eum Lee, and Choong Seon Hong*

Computer Science and Engineering, Kyung Hee University

{anselme, daeunlee, cshong}@khu.ac.kr

Abstract

Content Centric Networking is one of most important architectures in Future Internet for content sharing and retrieval, which consider content as central point of communication, where content is requested by name rather than its physical location address. In CCN, to request content, a consumer emits an Interest packet, which has a name that identifies the requested content. As soon as the Interest packet reaches a node that has the requested content, a content packet/ data chunk is sent back to the consumer through the Interest reverse path, which carries both the name and Data. When the Interest packet is transmitted, it is after receiving its corresponding data chunk that the next Interest requesting the next data chunk will be transmitted. The size of Interest packet is very small, and for the content with big size in terms of Megabytes, Gigabytes, Terabytes, etc. requires a consumer to send many numbers of Interest packets in one to one approach, in which may lead to link underutilization and unnecessary delay. To overcome the above highlighted problem related to content with big size, we propose an improved Interest forwarding for CCN, which is based on Interest packets grouping through an improved consumer sliding window, and fast recovery & fast retransmission. Our simulation results show that our proposal achieves high performance with higher throughput and lower delay over other existing similar proposal in the literatures.

1. Introduction

Since last many years, many research projects such as US NSF GENI, EU FIA, etc. were funded for focusing on definition of new Future Internet (FI) architecture [1]. Efficient, secure contents diffusion, and retrieval are the ones of the major points of the FI architecture. Contents Centric Network (CCN) proposal [2] is the one of the most important architectures for contents sharing and retrieval in FI, where content name is the central point of the communication as the replacement of its physical location address.

In CCN, contents are requested and retrieved by names rather than the IP addresses. There are two types of CCN 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. As described in figure 1, In CCN, there are three main data structures, namely Forwarding Information Base (FIB), Pending Interest Table (PIT) and Content Store (CS) or memory. To receive content, consumer sends out a content request called Interest packet, when a router receives Interest packet, it checks if the requested content/chunk is cached in CS. If the chunk packet is cached in CS, the router returns the chunk packet to the consumer in reverse path of Interest packet. If the chunk is not cached in CS, the router checks PIT, in which records unsatisfied Interests and faces in which Interests are coming from. If there is a similar entry in PIT, then the node appends Interest to the existing PIT entries. If there is no similar PIT entry, the new entry is created, and the router checks the FIB to determine where to forward Interest packet.

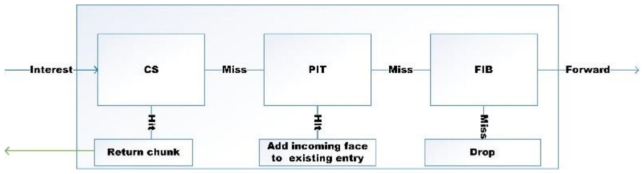

Figure 1: CCN data structures and forwarding

Each chunk packet is a segment of the requested content. One Interest packet is emitted to request one chunk or segment of data using one to one approach. When the Interest packet is transmitted, it is after receiving its corresponding chunk that the next Interest requesting the next chuck will be transmitted. The size of Interest packet is very small, and for the content with big size in terms of Megabytes, Gigabytes, Terabytes, etc. requires a consumer to emit many numbers of Interest packets in one to one approach, in which may lead to link capacity underutilization and unnecessary delay [3]. To overcome the above highlighted problem related to content with big size, we propose an improved Interest forwarding for CCN, which is based on Interest grouping through an improved consumer sliding window, and fast recovery & fast retransmission. We have simulated our proposal and compared it with existing CCN sliding window proposal by using ndnSIM 2.0 [4, 5]. The simulation results show that our proposal achieves high performance with higher throughput and lower delay over other existing similar proposal in the literatures.

The rest of the paper is structured as follows, Section 2 describes in details our Improved Interest Forwarding for CCN, while Section 3 provides performance evaluation of our proposal, which includes experimental setup and results. We conclude our paper with future scope and conclusion in section 4.

2. An Improved Interest Forwarding for CCN

In CCN, packet is forwarded based on its name; i.e. on every incoming Interest packet, the router performs its name lookup. If the chunk is cached in CS, the router returns the chunk packet to the consumer in reverse path of Interest packet. Otherwise router decides on how, and where to forward the packets through the use the forwarding strategy.

In CCN, when data chunk is discovered, it flows down to the receiver in Interest reverse path, where router on transmission path can cache returning chunks for serving similar requests in the near future from other consumers. Due to the caching functionality, data chunks keep changing their locations, it becomes harder to calculate the exact Round Trip Time (RTT), and to monitor end to end communication between consumer node and source of the content. Returning chunk acts as the acknowledgment of received Interest, and based on which consumer node keeps updating RTT and timeout values.

In figure 2, we define RTT as a length of time of sending Interest packet i at time T and receiving correspondent chunk i at time R.

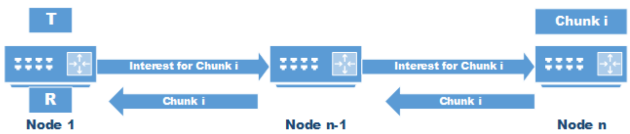

Figure 2: RTT

In our Improved Interest Forwarding for CCN proposal, we assume that there is no consumer, which requests only one data chunk; always consumer request the content partitioned into many chunks. From this assumption, there is a possibility of finding consecutive data chunks cached on adjacent or nearest nodes, and this motivates us to borrow some ideas from TCP's Retransmission Timer [6] and modify them for improving CCN Interest Forwarding.

For Interest forwarding, sliding window defines the number of Interests that can be sent without waiting corresponding chunk. In this paper, our sliding window starts with ten Interest packets as initial sliding window size, i.e. consumer node starts initially sending a group of ten Interest packets. On each chunk received, the initial number of Interest packets gets increased by one Interest packet. On timeout, consumer node restarts the process with ten Interest packets per group and double timeout period.

In CCN, users send Interest packets to request the content they need. Each Interest packet has a lifetime, and when the lifetime expires, the Interest gets removed from PIT and the consumer needs to retransmit unsatisfied Interest. For the content requested through sliding window approach, the consumer needs to retransmit the unsatisfied Interest (s) after receiving last chunk requested in sliding window. This may cause the link to become idle for a short interval of time between current sliding window and next sliding window.

To overcome the above issue, for fast recovery and faster retransmission, on each chunk received, the node computes the RTT, Smoothed Round Trip Time (SRTT), Round Trip Time Variation (RTTVAR) and Faster Retransmission Time (FRT) and retransmission timeout (RTO). More details about RTT, SRTT, and RTTVAR are well described in [6]. We consider timeout (RTO) as a time length that a node needs to wait until it concludes that the submitted Interest has failed to return a corresponding data chunk. Faster Retransmission Time (FRT) introduced in this paper, helps the router to detect unsatisfied Interest from large sliding windows where pending Interest packets are more than twenty Interest packets, and retransmit it without waiting RTO.

$$\mathrm{SRTT} = \mathrm{RTT} \quad (1)$$

$$\mathrm{RTTVAR} == \mathrm{SRTT}/2 \quad (2)$$

$$\mathrm{RTTVAR} = \mathrm{SRTT} + (K * \mathrm{RTTVAR}) \quad (3)$$

$$\mathrm{FRTO} = \mathrm{SRTT} + \mathrm{RTTVAR} \quad (4)$$

Where K is strength parameter. For each submit Interest packet, router records sequence number, keeps tracking sequence number, and the number of pending Interests waiting corresponding chunks.

For fast recovery and fast retransmission conditions, we introduce other two variables, namely retransm and FR (Fast Recovery), where

$$\mathrm{retransm} = (\mathrm{m\_seqMax} - \mathrm{current\_seq})/2 \quad (5)$$

$$\mathrm{FR} = \mathrm{sequenceNumber} - \mathrm{retransm} \quad (6)$$

For $\mathrm{retransm} > 10 \; \&\& \; \mathrm{FRTO} < \mathrm{RTO}$, the router retransmits Interest that has a sequence number which is less that FR and update RTO.

3. Performance evaluation

In this section, we evaluate the performance of our Improved Interest Forwarding for Content Centric Networking through simulations. In our simulations, ndnSIM 2.0, an NS-3 based NDN simulator [4, 5] is used.

3.1 Simulation Scenarios

In our simulation scenario described in figure 3, we have two consumers, in which consumer 1 and consumer 2 request one part of content with names /fi/presentation/part1 and /fi/presentation/part2 respectively. Both parts of the

presentation are hosted on separated Content Provider's servers, where each part has the size of 2 GB, while payload size is 1024. Cache size is set to 10000 chunks, while initial Interest and RTO lifetime are 4 seconds and queue size is 100 packets. Initial consumer window is 10 Interest packets, while Initial RTT is set to 0.01 seconds and K was set to 4.
During 50 seconds of simulation, each link has a capacity 2000 Mbps of a bandwidth, and 10ms of propagation delay. For Interest forwarding, we have used broadcast as forwarding strategy.

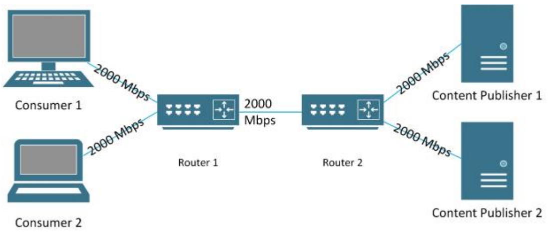

Figure 3: Simulation Scenario

3.2 Simulation Results

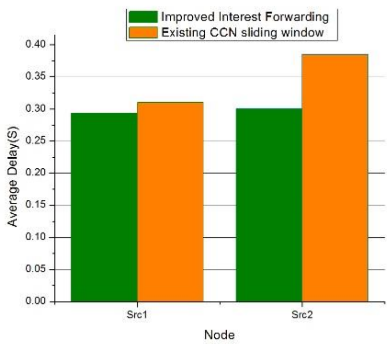

Figure 4: Average delay comparison

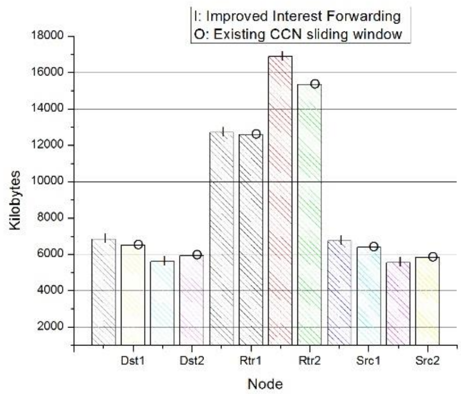

Figure 5: Throughput comparison

Our experiment results presented in figures 4, 5 and 6 show that our Improved Interest Forwarding for Content Centric Networking has high performance (delay and throughput) than the existing CCN sliding window proposed in [4, 5].

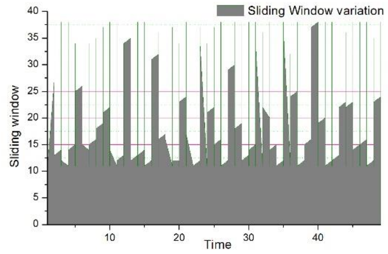

Figure 6: Our proposal sliding window variation

4. Future Scope and Conclusion

In comparison to existing CCN sliding window proposed in [4, 5], our experiment results show that our proposal reduces delay and increase throughput, but given to the complexity interaction between consumer driven transport and caching, a deep analysis with a mathematical model is needed to handle caching dynamicity in CCN forwarding. In the future, we plan to extend our analysis in modeling the interaction between CCN forwarding and caching.

5. Acknowledgement

This work was supported by the Industrial Core Technology Development Program (10049079, Development of Mining core technology exploiting personal big data) funded by the Ministry of Trade, Industry and Energy (MOTIE, Korea) *Dr. CS Hong is the corresponding author.

6. References

[1]. Carofiglio, Giovanna, et al. "Modeling data transfer in content-centric networking." Proceedings of the 23rd international teletraffic congress. International Teletraffic Congress, 2011.

[2]. Jacobson, Van, et al. "Networking named content." Proceedings of the 5th international conference on Emerging networking experiments and technologies. ACM, 2009.

[3]. Byun, Dojun, Byoung-Joon Lee, and Myeong-Wuk Jang. "Adaptive flow control via Interest aggregation in CCN." Communications (ICC), 2013 IEEE International Conference on. IEEE, 2013.

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

[5]. A. Afanasyev, I. Moiseenko, and L. Zhang, "ndnSIM: NDN simulator for NS-3," NDN, Technical Report NDN-0005, 2012

[6]. RFC6298, V. Paxson, et al. "Computing TCP's Retransmission Timer." (2011).