An Effective Congestion Control Scheme in Content-Centric Networking

Sarveswara Reddy.V
Department Of Computer Science
Saveetha University

Abstract— Content-Centric Networking (CCN), a typical future Internet architecture focused on content dissemination and re-trieval, brings a paradigm shift in network model by addressing named-data instead of host location. Many fields related to CCN become the research hotspots, such as caching and name-based routing. However, little work has addressed on its transmission protocol in literature. In this paper, we proposed an effective Congestion Control Scheme (CCS) in CCN through combining a router-driven congestion forecast with a requester-driven Interest control protocol. Each router detects the network congestion status and then informs the requester by setting Congestion Information Bits into the returned Data packet passed by. Then the requester dynamically adjusts its Interest sending window according to the received feedback congestion information. Fur-thermore, in our scheme, a Fair Queuing algorithm is used on each router, allowing max-min fair share of the link capacity. A performance evaluation using simulation designed on OMNet++ network simulator showed that the proposed CCS causes fewer packet losses, decreases Data packet delay, treats each flow fairly and maximizes the use of bandwidth.

I. Introduction

Internet has involved significantly in the past years, and today it is generally recognized that the future Internet should pay more attention on content dissemination and retrieval instead of end-to-end communications. Following this idea, some new Internet architectures have been proposed, such as PSIRP , DONA , CoNET and CCN [1]. Among them, Content-Centric Networking (CCN), pioneered by Van Jacobson, has recently gained momentum in the research community and originated.

Previous works on CCN have mainly focused on the global architecture design [2] [3], while less effort has been devoted to the study of the transport mechanism and congestion control in such architectures and notably, on the definition of a CCN flow control protocol. Paper [4] proposed an Interest Control Protocol in CCN, It uses TCP-like slow start and retransmit-overtime mechanism. However, the RTT for each packet varies greatly as different Interest may be satisfied by different node on the transmission path, it leads to the difficulty in setting the timeout parameter τ . On the other hand, this protocol doesn't provide any fairness scheme for different flows. In CCN, a strategy layer is reserved to provide traffic control by adjusting the number of unsatisfied Interest allowed. At an end node, the strategy component decides when to retransmit an unsatisfied Interest and how often to send out the following Interest.

Unfortunately, there is little work addressed the trivial traffic control design at this strategy layer in literature until now. The objective of this paper is to try to give some appropriate specification on congestion control at CCN strategy layer.

In this paper, we propose a Congestion Control Scheme (CCS) in CCN. The requester detects network congestion situation by checking the information carried in the Congestion Information Bits (CIB) in the returned Data packet, which is set by the routers, instead of by any timeout event or packet loss event. The router serves for each flow in its Data buffer fairly and computes the congestion degree for each flow by detecting the queue size of this flow. After the requester receives the Data packet, it changes its Interest sending window according to the feedback CIB. In this case, few packets will be discarded and all flows share the bandwidth equally, and the bandwidth is used sufficiently. The contribution of this paper is threefold:

1. We propose an effective congestion control scheme through adjusting the Interest sending window at end node, inspired by congestion forecast at the intermediate router. At the best of our knowledge, this is the first scheme proposed by considering both end node and network routers participation together.
2. A simple fair queuing algorithm is used on each router, allowing max-min fair share of the link capacity for each flow.
3. We develop a CCN prototype with OMNET++ simulator, and the result showed that the proposed CCS causes fewer packet losses, decreases Data packet delay, treats each flow fairly and maximizes the use of bandwidth.

The remainder of the paper is organized as follows: In Section II, we give a brief description of CCN node model and the scheme objective. Then we explain our CCS in detail in Section III. After that, We show the performance evaluation in Section IV. At last, we conclude our work in Section V.

II. Model Description And Scheme Objective

A. CCN Node Model

There are two types of packet chunk in CCN, Interest and Data packet. For the sake of the simplicity, we do not carefully distinguish packet and chunk (transmission unit in CCN) in this paper, although a chunk is always larger than a packet. A Data chunk is transmitted only in response to an Interest with the same chunk name. A CCN node consists of three main parts: Content Store, Pending Interest Table (PIT) and Forwarding Interest Base (FIB). The Content Store is a large repository used to cache Data passed by it, leading to the request chunk be found at the intermediate router instead of the remote end server. The FIB is used to route the Interest to the original publisher and the PIT is used to keep the incoming interface information of the Interests which are not satisfied yet.

B. Scheme objective

According to the CCN characteristics and the fast increased demand for high-speed Internet, it is advisable to design a CCS with the following objectives, which is used to guide our CCS design. (1) First and foremost, the CCN router should try to avoid packet loss and Interest retransmission which wastes the network resource and increases the network load. To gain this goal, congestion forecast in advance is necessary in order to avoid the real congestion occurrence. (2) Second, it is preferable if the CCS could decrease the delay, especially for the low data rate flows. (3) Third, the CCS should ensure the fairness among flows. In this paper, we define it as each flows fairness which means that every flow will share the bandwidth equally and low data rate flow gets low delay. (4) The last objective is to maximize the bandwidth utilization in order to increase the whole throughput of the network.

III. CCS

In CCS, the router serves for each flow at each interface with Per-flow Fair Queueing (PFFQ) Algorithm instead of First Come First Service (FCFS) algorithm. Then, two bit fields are added into the CCN Data packet, called Congestion Information Bits (CIB). Before a returned Data chunk is sent back to the requester, the router detects the congestion degree for this flow and set the CIB in the Data packet. When the end requester receives the Data packet, it dynamically adjusts the Interest sending window (*Iswnd*) of Interest according to the CIB in the Data packet. As a result, the queue size of each flow would keep stable and each flow share a fair bandwidth. We explain it separately in detail:

A. *CIB and explicit notification*

In CCN, congestion is defined as the overflow of the queue at an interface caused by too many Data chunks swarming into the CCN router. Thus the router could detect congestion degree according to the number of Data packets in each flow. In CCS, routers explicitly notify the congestion degree to the requester by the CIB in each returned Data chunk. Two bits in CIB indicate four kinds of congestion degree: “Excellent” (00), “Good” (01), “Bad” (10) and “Awful” (11). “Excellent” means the link is idle so the requester could increase its *Iswnd*. While “Good” means the congestion degree is appropriate, so the requester should keep its *Iswnd* unchanged. In “Bad” and “Awful” situation, the requester should decrease its *Iswnd*. In this way, congestion degree would be explicitly notified to the requester in time so that the requester could take actions to avoid the real congestion occurrence.

B. *PFFQ in router*

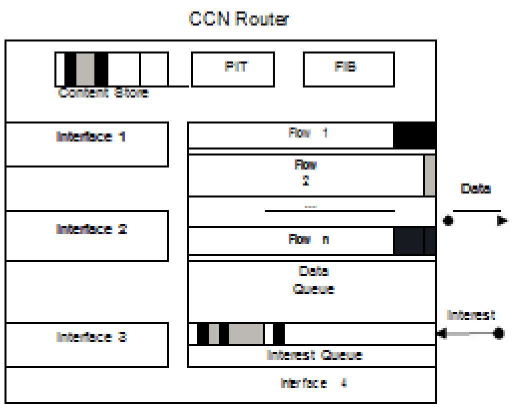

Fig. 1: Per-flow queue in CCN router.

PFFQ means that each flow shares the bandwidth equally. Especially, it is necessary to ensure low delay for low data rate flow. As presented in Fig. 1, a Data queue for each flow is set up in the interface’s Data buffer. These queues are logical queues which means that all queues may be in one physical buffer. The interface sends out Data chunks by picking from each queue circulatory so that each flow has the same probability to be sent out.

C. *Congestion detection in router*

As stated above, the CIB in the returned Data chunk is used to inform the congestion degree to the end requester, so it is crucial to set the appropriate CIB value. Here we describe the algorithm how to set the CIB for each flow. All the values and notations used in our algorithm are given in Table I.

It is noticed that we assume that Interest wouldn’t result in congestion, as Interest packet size is much smaller than Data packet size. All the values and notations used in our algorithm are given in Table I. Assume that there are *F* flows

Table I Notations

B	Total buffer size of Data queue.
F	Number of flows in the Data queue.
Q	Queue size of flow in the Data queue.
W_q	A weight factor. $0 < W_q < 1$.
avgQ	Average queue size of flow.
maxQ	Threshold of maximum queue size of flow.
minQ	Threshold of minimum queue size of flow.

in the interface now, each flow share a buffer with size BF . Firstly, before the router sends out a Data chunk, it computes the average queue size (avgQ) of the flow which the Data belongs to by Eq. (1), where Wq is a weight factor.

$avgQ = (1 - Wq) \cdot Q + Wq \cdot Q$ (1)

If the avgQ ≤ minQ, the CIB will be set to “00” which means no congestion on the interface. While if the avgQ ≥ maxQ, the CIB will be set to “11” to mean the heavily congestion.

If minQ ≤ avgQ ≤ maxQ, CIB may be set to a value among “00”,“01” and “10”, representing different degree of congestion respectively. Here we borrow the idea of RED (random early detection) [5], where the packet is discarded in a probability which is linear with the AvgQ. The discard probability P is calculated by Eq. 2. Therefore, the CIB be set to “00” with a probability (1 − P), while set to “01” or “10” with a probability P . Specifically, in the latter case, in order to accurately react the congestion degree, the CIB is set to “01” if AvgQ ≤ (MaxQ + MinQ) • 50%, otherwise it will be set to “10” as shown in Eq. 3.

P = avgQ − minQ (2) maxQ − minQ

CIB = “00”, (1 − P)
“01”or“10”, P (3)

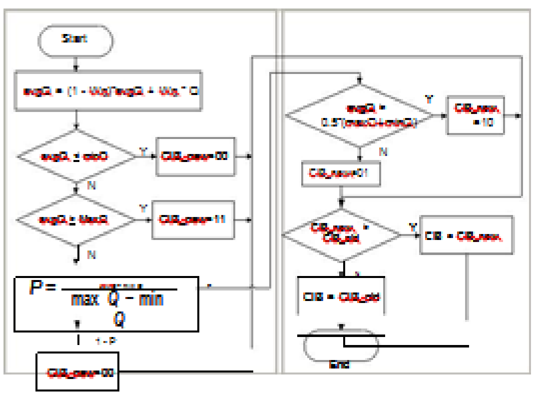

Fig. 2: Algorithm for generating CIB.

In addition, another important principle on setting CIB is that one CCN router can't change the CIB to a better situation than that indicated in the Data chunk entering the face from last router. If a CIB in the incoming Data packet is "10" (set by last router), it can't be changed to "01" even though the calculation result of this router is "01". The algorithm is presented in Fig. 2.

D. Interest sending window adjustment

As we know, the fundamental method to avoid congestion is to control the amount of the packets that enter the network. In CCN, one Interest chunk sending to the network is bound to cause a Data chunk return to the requester through one or more links. Therefore, it is significant to adjust the Iswnd dynamically according to the network congestion situation, just as cwnd in TCP. Here, Iswnd in CCS is defined as the maximum number of Interests that can be outstanding at any time. A threshold ssthresh is set for Iswnd to avoid the Iswnd increasing too fast after Iswnd is greater than ssthresh.

Since the four states of CIB can accurately predict the different congestion situation, it is preferable to slightly adjust the network traffic in order to maximize the bandwidth usage. In this paper, we adjust the Iswnd with different strategy when receiving different CIB value, in order to avoid the traffic fluctuation. The *Iswnd* is initialized by one chunk at the transmission beginning. When the requester receives a CIB of "00", the *Iswnd* increases by one chunk in case that *Iswnd* is less than *ssthresh*, otherwise it increases by 1*chunk/Iswnd*. If the requester receives a CIB of "01", it just remains the *Iswnd* unchanged. If the CIB is "10", the *Iswnd* decreases 0.25*chunk*, while if the CIB is "11", the *Iswnd* decreases by 0.5*chunk*. This algorithm is presented in Table II.

Table 2: Interest Sending Window Adjustment Algorithm

"00"	if(Iswnd < ssthresh) Iswnd = Iswnd old + 1
	if(Iswnd ≥ ssthresh) Iswnd = Iswnd old + 1chunk/Iswnd old
"01"	Iswnd = Iswnd old
"10"	Iswnd = Iswnd old − 0.25chunk
"11"	Iswnd = Iswnd old − 0.5chunk

IV. PERFORMANCE EVALUATION

The aim of this section is to analyze, through simulations, the performance of the CCS in CCN. We have developed the CCN module in OMNeT++ network simulation framework [6]. As far as we know, there is no CCN module in OMNet++ at the time of paper writing. We start out simulation with a

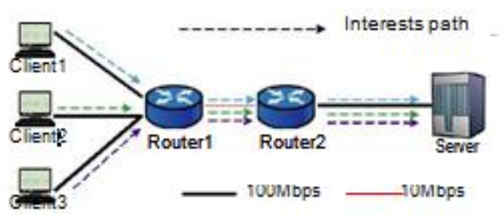

Fig. 3: A simple simulation scenario.

simple scenario presented in Fig. 3. Each of the three clients requests for a different content file with the same size of 10MB at different start time: client1 starts at the simulation beginning, client2 starts at 4s while client3 starts at 8s. The size of Data chunk is 4KB, so in order to get a 10MB file, client needs to send 2560 Interests at least. All the links have 5ms propagation delay and each link bandwidth is 100Mbps except the link between router1 and router2, which is 10Mbps. So router2 maybe congested by Data packets. In this simple model, all the Interests will be forwarded to the server as we didn't consider the router's cache function. All the parameters are: *ssthresh* = 128, Total buffer size of interface *B* = 500 chunks, *maxQ* = 350*chunks*, *minQ* = 150*chunks*. The simulation results are presented in Fig. 4, 5 and 6. The *Iswnd* of each client increases quickly to the suitable value and then keeps stable until a new flow is added to the router, as shown in the Fig. 4. In Fig. 5, when a new flow is added into the router, the interface's queue size of router 2 converges quickly and it is not out of buffer (The buffer size is 500 chunks), leading to less packet loss. Fig. 6 shows that

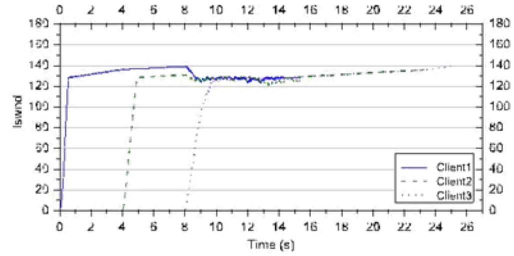

Fig. 4: Iswnd of each client.

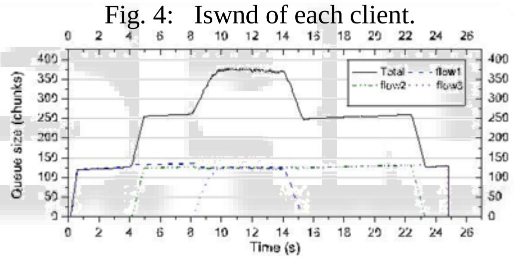

Fig. 5: Queue size of router2.

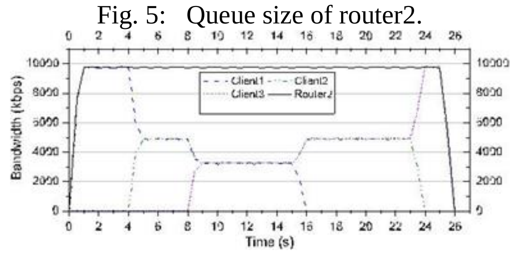

Fig. 6: Bandwidth consumption of three clients and router2.

the three clients share the bandwidth equally and the link's bandwidth is sufficiently used without fluctuation. All these figures evaluate the CCSs merits due to its congestion forecast mechanism and the Iswnd elaborate adjustment strategy. Then

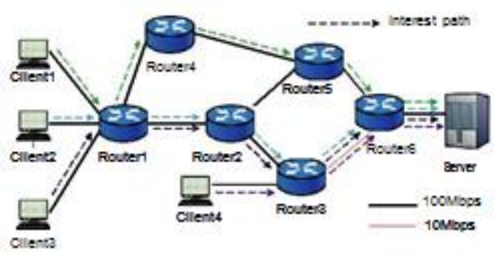

Fig. 7: A complex simulation scenario with multiple nodes.

we extend our simulation to a complex scenario with multiple nodes in an arbitrary topology as presented in

Fig. 7. In this scenario, all routers cache the Data chunks passed by it in order to serve the following request nearby. Here, the total cache size of all routers is about the 20% of the total content volume in the CCN network. All clients start sessions at the simulation beginning except client2, while the client2 requests the same content file downloaded by the client3 at 10s. The other parameters are same as that in the last scenario. In this simulation, we compare CCS with the original CCN which use FIFO queue in its interface buffer and fixed *Iswnd* (without any congestion control scheme), by checking file downloading time, retransmitted Interests number and average Data chunk delay. The simulation results, as presented in Table III, shows

Table 3: Simulation Results Of Multi-Node Scenario

scheme	download time(s)	re-Interests	packet delay(s)
CCS	12.5	0	246.3
CCN (Iswnd = 1)	279.2	0	36.5
CCN (Iswnd = 20)	13.64	0	105.2
CCN (Iswnd = 30)	9.94	2367	407.4
CCN (Iswnd = 50)	9.92	2527	425.9

that the proposed CCS outperform the original CCN in that it can adjust its *Iswnd* to appropriate value and achieve balance between file downloading time, retransmitted Interests number and Data packet delay.

V. Conclusion

In this paper, we presented CCS, an effective congestion control scheme designed to avoid the congestion in CCN network. With the congestion forecast information in CIB, which is set by the CCN routers according to each flow's queue size in the output buffer, the requester adjusts its *Iswnd* dynamically and elaborately in order to avoid packet loss and request retransmission. A Fair Queuing algorithm is used on each router, allowing max-min fair share of the link capacity. We demonstrated analytically and by simulation that CCS could achieve few packets loss, lower packet delay and high bandwidth usage.

