Congestion Control for Content-Centric Networking Based on Protocol-Oblivious Forwarding

Ying XIA, Lei WANG, Fang-jie HOU and Yi-ran WANG

Department of Automation, University of Science and Technology of China, Hefei, China

Keywords: Content-Centric networking, Software-Defined networking, Protocol-Oblivious forwarding, Congestion Control.

Abstract. The traditional TCP/IP network architecture has been already unable to meet the growing demand of content distribution. The combination of CCN (Content-Centric Networking) and SDN (Software-Defined Networking) has been gradually considered as the next network architecture. In face of the lack of end to end session in CCN, the design of a congestion control mechanism becomes very important. Based on the new SDN protocol—POF (Protocol-Oblivious Forwarding), we propose a congestion judgment strategy based on explicit feedback, with different values representing different network congestion situations. In addition, the data receivers and intermediate nodes will work together for congestion control. The proposed strategy is evaluated by experiments. The experiments results show that this scheme works well and can effectively control congestion in POF-CCN.

Introduction

CCN (Content-Centric Network) [1] has changed the original host-host model in IP. Contents are retrieved directly by their names. This architecture has the advantages of simple structure and good expansibility. However, CCN has lack of global recognition for the whole network, which leads to unreasonable resource allocation. With the emergence of Software-Defined Networking (SDN) [2], the network layer protocol can obtain a greater degree of freedom. SDN separates control plane from data plane, and makes it easy to configure the network. Protocol-Oblivious Forwarding (POF) [3], as a key enabler for highly flexible and programmable SDN, can support any upper layer self-defined protocols. Accordingly, we propose a POF-CCN network architecture.

With the growth of content, congestion control becomes an indispensable role. Because there is no connection status in CCN and returned data packets which are always come from multiple sources, in which network congestion is inevitable. In POF, we propose a congestion judgment strategy based on explicit feedback. In addition, the receivers and intermediate nodes will work together for congestion control.

Related Work

In this section, we discuss and analyze current research about congestion control in CCN. ICP in [4] and ICTP in [5] propose congestion control strategies by predicting RTT. ICTP is similar to TCP congestion control and ICP ignores the fact that returned data packets are always from multiple sources. HoBHIS in [6] adjusts packet transmission rate at intermediate routing nodes by the bandwidth, buffer size and other relevant parameters. This is a congestion control method hop-by-hop. CCTCP [7] improves the accuracy of RTT estimation by the feedback of each hop, but the cost of maintaining multiple congestion windows and timeout values for each flow is larger. Fu T et al [8] uses explicit feedback. In order to view the congestion status, it sets the CIB (Information Bits Congestion) in the return packet. The Interest adjusts the send window according to the CIB, which is determined by the intermediate router. This method takes full account of fairness, but it ignores the influence of interest packets. Tang Xiao et al [9] proposes to divide the link state into three forms (Free, Busy and Congestion) according to the average length of the transmission queue.
It uses the special interest packet—InterestNack. The scheme ignores the performance of the intermediate nodes.

**Congestion Judgment Strategy Based on Explicit Feedback**

**POF-CCN Architecture**

The global view of our POF-CCN system architecture is shown in Figure 1. The top is control layer with one or several POF controllers attached with a server for a better performance. With the POF controller’s centralized control, we can monitor the status of network. The middle layer is composed of POF switches and service nodes. Every service node is attached to a POF switch. POF switch and its service nodes work together to realize a CCN Router’s functions. The service node takes the routing and caching function of CCN. The third layer is composed of users attached to the switches.

![Figure 1. POF-CCN system architecture.](image1)

![Figure 2. A simple router model.](image2)

**Congestion Judgment Strategy Based on Explicit Feedback**

In CCN, we set flag in the interest packet instead of the returned data. When the congestion is detected, the feedback will immediately been brought back to the receivers. In addition, we also detect congestion in intermediate nodes. A simple router model is illustrated in Figure 2. A router has lots of interfaces, and each interface corresponds to a transmission buffer. We only consider the flow in one direction. The number of interest packets from router R1 to R2 indicate the number of data packets to be returned to R1. So we calculate the queue length of interest packets in router R1 to judge the degree of congestion. The average queue length $avgQ$ is expressed as follows, where $Q$ is the actual queue length that is expressed as the average queue length using exponentially weighted moving average model (EWMA). $W$ is the weight. Symbol description of the algorithm is shown in Table 1.

$$avgQ = (1 - W_q) * avgQ + W_q * Q$$  \hspace{1cm} (1)

The nodes compare the average queue length with the set parameter and use the comparison results to change the flag. Slow start threshold (ssthresh) is used to avoid the congestion window growing too fast. These are three key points in our strategy: minQ, busyQ and maxQ. The queue length is divided into 4 parts in the coordinate axes according to these three values. At the beginning of the network transmission, the node detect the congestion and set the flag according to the interest queue. The congestion value on node should be compared with the value which is flagged in interest packet in order to realize real-time update. Calculation algorithm of congestion value is shown in Table 2.
A Joint Congestion Control Scheme

In order to observe the network state and prevent the congestion, we propose a joint congestion control scheme (CVUnion) in which data receivers and intermediate nodes work together for congestion control. Figure 3 shows the joint congestion control model.

Table 1. Symbol description.

<table>
<thead>
<tr>
<th>symbol</th>
<th>Symbol description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q</td>
<td>The queue length of interest packets.</td>
</tr>
<tr>
<td>W_o</td>
<td>weight, 0 &lt; W_o &lt; 1.</td>
</tr>
<tr>
<td>avgQ</td>
<td>average queue length.</td>
</tr>
<tr>
<td>maxQ</td>
<td>Maximum transmission queue length.</td>
</tr>
<tr>
<td>minQ</td>
<td>Minimum transmission queue length.</td>
</tr>
<tr>
<td>busyQ</td>
<td>critical value between idle and busy. minQ−busyQ=maxQ.</td>
</tr>
<tr>
<td>ssthresh</td>
<td>slow start threshold.</td>
</tr>
</tbody>
</table>

Table 2. Calculation algorithm of congestion value.

```
BEGIN
1 use (i) to calculate the average queue length avgQ.
2 CV_old = CV; // store original CV.
3 if 0 ≤ avgQ < min Q, CV_new = 00; // network free.
4 else if min Q ≤ avgQ < busyQ, CV_new = 01; // Network mild busy
5 else if busyQ ≤ avgQ < max Q, CV_new = 10; // network serious busy
6 else if avgQ ≥ max Q, CV_new = 11; // network congestion.
7 if CV_new > CV_old, CV=CV_new; // use the current value to update.
8 if CV > 00, feedback the congestion value to the receiver.
END.
```

Congestion Control at Receivers

An explicit feedback congestion control mechanism implemented by receivers. The receiver
regulate-rate algorithm is shown in Table 3, where cwnd (congestion window) indicates the size of send window.

**Congestion Control at Intermediate Nodes**

Intermediate nodes determine whether they should be shaped and make fair shaping to the flow if necessary. In order to better regulate congestion, we propose a scheme that intermediate nodes can cooperate with receivers to realize the fair bandwidth allocation and the interest queue shaping. In this scheme, we still use the interest queue length to judge congestion. When there is no congestion (avgQ < maxQ), we use max-min fair algorithm to allocate the bandwidth. Firstly, network bandwidth is allocated equally. The flow with minimum bandwidth requirement need to be processed sequentially. If the allocated bandwidth exceeds actual demand, the excess bandwidth will be allocated to the remaining flow. The bj means the actual allocated bandwidth of the jth flow, while Bj represents the actual bandwidth demand. This process is repeated until a flow’s allocated bandwidth is less than the actual demand, or all flows has been dealt with. When congestion appears (avgQ > maxQ, the tagged congestion value CV is ‘11’), the interest queue should be shaped. At each stage of the algorithm, congestion value and queue length has been updated. If delay list had kept the delayed request, the node should send it to the network.

**Experimental Evaluation**

The verification experimental topology is shown in Figure 4. Three clients request three different contents, and the size of the content is 10MB. The client 1, 2, 3 starts to send requests sequentially from 0s, 20s, 40s, and the minimum service bandwidth required for the three requests is 2 Mbps, 2 Mbps and 5 Mbps respectively. The size of the content chunk for each request is 4KB, so the 10MB file needs to send at least 2560 Interest requests. The link bandwidth between Router1 and Router2 is 10Mbps and the other link bandwidth is 100Mbps, so Router2 may send congestion. In this experiment, the router will not consider caching data packets. All Interest packet requests will be forwarded to the server. Other parameters are set as follows: ssthresh=250, buffer size for each interface B=500 chunks, maxQ=300 chunks, minQ=50 chunks, busyQ=150 chunks.

The comparison experimental topology is shown in Figure 5. We send 4000 requests for content chunks, and the size of each chunk is 4KB. These chunks are randomly distributed over two servers. The client 1, 2, 3 start to send requests sequentially from 0s, 20s, 40s. Other parameters are set as follows: ssthresh=250, maxQ=300 chunks, minQ=50 chunks, busyQ=150 chunks.

Figure 6 shows the changes of request window size. After the sending request of client 1 the window increase quickly to the slow start threshold. The interest packets are send gently until the client 2 joins. Network becomes busy, then client 1 reduces the window. Clients 2 synchronizes with client1 after initial increase and adjustment. After the request of client 3, the windows of client 1 and 2 are both reduced. After the quickly increase, client 3 enters the gentle phase. Because of the ssthresh limit and network feedback client3 reduces to a smooth level. Finally the three clients are stable. The proportion of the three windows is also consistent with Initial bandwidth demand ratio.

Figure 7 shows the bandwidth usage. The usage remains above 80% when there is only a client1. After the request of client 2, two clients share the same bandwidth. Adding the client 3, then the usage of client 1 and 2 are both reduced. When the request of client 1 is over, the network remains
the flows of client 2 and 3. The link bandwidth between Router1 and Router2 can meet the bandwidth requirements of them, so their bandwidth usage is similar.

From the experimental results we can see that the send window at receivers can be adjusted according to the network state, and the intermediate nodes can allocate bandwidth fairly according to the needs of users.

Figure 8 shows the comparable results. The windows of our joint scheme (CVUnion) and CHoPCoP are higher than ICP’s. In this experiment, the request of client 1 has been finished before the beginning of client 3. In our scheme CVUnion, the window enters the gentle phase after the quickly growth. However in the CHoPCoP, the window of client 3 experiences continuous rise and fall because of the delay of data packet feedback. The overall average bandwidth utilization of the three schemes is shown in Table 4. Our scheme has the highest utilization.

**Conclusion and Future Work**

The paper proposes a congestion judgment strategy based on explicit feedback, with different values representing different network congestion situations. In addition, the data receivers and intermediate nodes will work together for congestion control. Our scheme use the interest queue length to judge congestion and send the network state to receivers. Based on the explicit feedback, receivers adjust the size of send window by using method similar to AIMD. And the intermediate nodes can allocate bandwidth fairly according to the needs of users. The experiments results show that this scheme works well and can effectively control congestion in POF-CCN. As future work, we intend to ensure the effectiveness of the explicit feedback and combine the caching with our joint congestion control strategy.
Acknowledgement

This work has been supported by the "Strategic Priority Research Program" of the Chinese Academy of Sciences (Grant No. XDA06011202).

References


