

# Congestion Control Model Based on Hop-to-hop Feedback for Active Networks

Nie Fei,<sup>†</sup> and Li Zeng-zhi<sup>††</sup>

Xi'an Jiao Tong University, Institute of Computer Architecture and Networks, Xi'an, China

## Summary

Traditional end-to-end congestion control mechanism is unfit for active networks. A hop-to-hop feedback congestion control mechanism suitable for active networks is proposed in the paper. In an active node, all arriving active packets are firstly classified into queues according to the next node through which they will pass. The length of queues in buffer indicates congestion degree of a node. The congestion status of current node can be improved by sending control packet to its previous node. The correlative nodes will cooperate to control congestion of network. Theoretic analysis and simulation experiment express that the solution can eliminate congestion of network effectively and rapidly.

## Key words:

*congestion control; buffer; queue; hop-to-hop feedback.*

## Introduction

In contrast with traditional packet-switching networks, active networks[1] are programmable and can perform customized computations on the messages flowing through them, which improves the flexibility of network greatly. Unlike mid-nodes of traditional networks can merely store and forward packets, mid-nodes of active networks are provided with ability of computation. With the increase of arriving packets, active nodes are faced with much more pressure of computation, and congestion will occur more frequently in active nodes.

Traditional end-to-end congestion control mechanism don't fit for active networks, the reason is as below: traditional networks often use packet drop as an indication of network congestion, as the content of a packet remains unchanged in transmission, retransmitting lost packets is feasible. However, an active packet may have been computed by active nodes for several times when discarded, it costs much to discard such packet.

Only in gateways, at the convergence of flows, is there enough information to control sharing and fair allocation. Thus, we view the gateway congestion detection algorithm as the next big step. The goal of this algorithm is to send a

signal to the end nodes as early as possible [2]. The above is Van Jacobson's viewpoint about congestion control. In active networks, gateways are replaced by active nodes. As active nodes have ability of computing, they can collect and analyze useful information within active networks. Moreover active nodes can produce packets, through which control strategies would be sent to corresponding nodes. There already have been some studies on congestion control with active networking. One example is about active nodes using dynamic routing which aims to allow fluidity and route allocation and which ensures that networks resources are used efficiently[3]. The other is about Active Congestion Control (ACC) applying active networking to feedback congestion control on a high bandwidth-delay product network, shortening the feedback loop by filtering traffic in the network near congestion[4]. These researches improve congestion performance in some ways and give us some illumination.

Our research works focus on congestion control for active networks. The concept of queue length matrix is defined to manage inner information about active nodes. Our model consists of active packets and active nodes. The control strategy of active nodes comes from analysis of inner information in active nodes. This paper is organized as follows: in Section 2, the congestion control model is presented; In Section 3, the principle of hop-to-hop feedback control is introduced; In Section 4, the validity of the model is discussed by a simulation experiment.

## 2. Congestion Control Model

As mid-nodes perform some computations on packets passing by, mid-nodes will spend more time to process packets in active networks than in traditional networks. We analyze computation delay is one of the main reasons to cause congestion in active networks. To simplify research, we assume packets computed by active nodes will be transmitted in time, and bandwidth is not the main factor causing congestion. So we concentrate our research on active packets and active nodes.

---

Manuscript received March 25, 2006.

Manuscript revised March 30, 2006.

## 2.1 Active Packet Model

Fig. 1-a shows us a basic security active packet format[5], to simplify research, we assume that the routing path of a packet is decided when it is injected into an active network. Beside source address and destination address, an active packet should include the address of each mid-node through which it will pass. In our packet model, the information is saved in the field of 'Mid-node Chain'(Fig. 1-b).

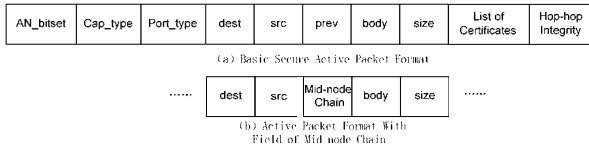


Fig. 1 Active Packet Model

As congestion node should send control information to its previous node, we define a control packet model which includes the field of control policy.

## 2.2 Active Node Model

As shown in Fig. 2, an active node is composed of a classifier, buffer queues, a queue manager, computing units and a computing strategy controller.

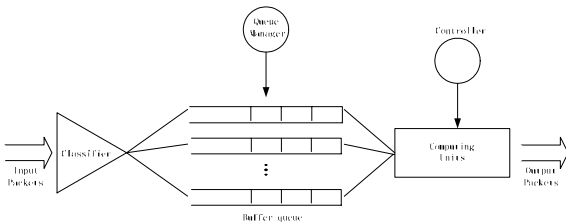


Fig. 2 Active Node Model

Each component of an active node is explained as follows:

**Classifier**, when a packet arrives a node, it will be classified by the classifier and be sent to a buffer queue according to next node in its field of 'Node Chain'.

**Buffer queue**, in our model, the strategy of congestion avoidance and congestion control is decided by buffer queues. The quantity of buffer queues in an active node is equals to the quantity of neighbor nodes of the node. That is, we supply a buffer queue for every neighbor node of current node. Every passing packet will be classified to corresponding queue according to its next node. To guarantee impartiality, the policy of the queue is FIFO.

**Queue manager**, in charge of monitoring buffer queues and collecting information from queues, a queue length matrix (Fig. 3) is defined to manage status of the queues, the element  $a_{ij}$  of the matrix indicates current quantity of packets which come from node  $i$ , and would be routed to node  $j$ .

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,m} & a_{1,m+1} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,m} & a_{2,m+1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,m} & a_{m,m+1} \end{bmatrix}$$

Fig. 3 Queue Length Matrix

**Computing units**, being computing resources of an active node, each unit is an integrated executing environment. Packets in the queue wait to receive services from Computing units. The computed packet will be transmitted to next node.

**Computing strategy controller**, in charge of computing resources allocation to each queue. According to received congestion control request from neighbor nodes, it decides computing resources allocation strategy.

By monitoring buffer queues, an active node performs congestion forecasting. Once the length of a queue exceeds predefined value, the active node considers there is potential congestion in the node, and then according to the queue length matrix, notifies correlative previous nodes by sending control packet. When the previous neighbor node receives the packet, its computing strategy controller will take step to slow down processing speed to corresponding queue. This will reduce congestion pressure to next node. In the same way, the congestion control information will be transmitted to its previous nodes, in the end, the control information will arrive nodes. The edge nodes will notify applications to reduce packet sending rate dynamically. Finally, congestion is eliminated. This is our principle of hop-to-hop feedback congestion control model (Fig. 4).

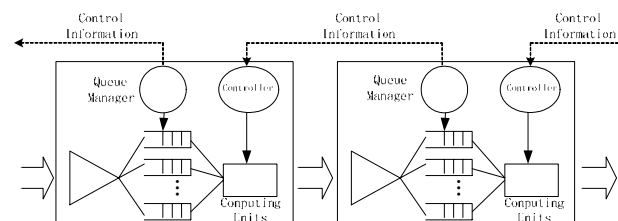


Fig. 4 Hop-to-hop Feedback Congestion Control Chain

### 3. Congestion Control Algorithm

For the convenience of simulation, packet length is limited to 1KB.

#### 3.1 Parameters of Describing a Queue

In our model, we define three parameters to describe a queue: queue length to indicate a node entering into congestion –  $Q_{con}$ ; queue length to indicate a node on light load status –  $Q_{min}$ .  $L$  is the buffer length allocated to a queue by a node.  $L$  should be a moderate value, too short, unfit for congestion, too long, queuing delay increasing accordingly.

#### 3.2 Congestion Control Algorithm at an Mid-node

The aim of congestion control at an active node is to improve efficiency of the network by keeping queue length in a proper range. When light congestion occurs, by controlling packet flow rate, the actual queue length should be adjusted to between  $Q_{con}$  and  $Q_{min}$  in time.

Control strategy is defined as follows:

When queue length of the node increases gradually beyond  $Q_{con}$ , supposing it's queue  $j$ , it implies there is a light congestion in the node. The node will produce and send a control packet to its previous node. The previous node is selected by inspecting the queue length matrix. Firstly, we select the column  $j$  by queue  $j$ , then we choose the biggest value from column  $j$ , if the row of the biggest value is  $i$ , the previous node is node  $i$ . Then the computing strategy controller of the previous node will reduce computing resources allocated to the queue corresponding with the congestion node to half of its current level.

When a queue length decreases gradually below  $Q_{min}$ , it implies the current node has ability to computing more packets. The node will produce and send a control packet to its previous node by inspecting the queue length matrix. Then the computing strategy controller of the previous node will increase computing resources allocated to the queue corresponding with the congestion node by half of its current level.

When queue length remains between  $Q_{con}$  and  $Q_{min}$ , the node works well and don't produce a control packet.

Theoretic analysis express that the algorithm has some characteristics. Firstly, it responds quickly, when there is light congestion in the node, the previous node of the congestion node will receive control information soon. Then it will reduce the number of packets transmitted to the congestion node. Secondly, the correlative nodes form

a chain from congestion node to edge node, and will cooperate to control congestion of networks. The buffers of the node form a buffer pool, which will contain burst flow effectively. Only the burst flow remains long time, congestion control request will be transmitted to edge node, the edge node will notify applications to reduce packet sending rate to eliminate congestion.

#### 3.3 Congestion Control Algorithm at an Edge Node

Edge nodes connect with host, in our model, when congestion control request arrives the edge node, the edge node will notify applications in the host to change packet sending rate.

### 4. Simulation Experiment

In order to analyze system performance, we set up a simulation model(Fig. 5). Node A and node B act as edge nodes, node C、node D and node E act as mid nodes, data flow from node A will be transmitted to node E via node C, data flow from node B will be transmitted to node D via node C.

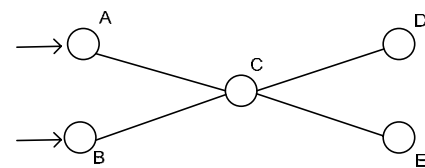


Fig. 5 Simulation Model Topology

As data flow from node A and data flow from node B converge at node C, congestion maybe occurs in node C. Then node C will send control packet to node A or node B to adjust data flow rate towards node C.

Now, we inspect our analysis by a simple simulation experiment. Supposing original transmitting rate of node A is one packet every 5 milliseconds, that of node B is one packet every 10 milliseconds, it cost 15 milliseconds for node C to compute and transmit a packet, computation resources are fairly allocated to each queue in node C. The buffer length of each queue is set to 100,  $Q_{con}$  is set to 50,  $Q_{min}$  is set to 20. Node A and node B have sent packets for 6 minutes.

Having done the experiment for 100 times, we have the conclude that after 642 to 1990 milliseconds, the sending rate of node A and node B is adjusted to nearby 29 milliseconds per packet, the fastest is 22, the slowest is 38, the value of 29 occurs most frequently. After the node

arriving balance, the max queue length of packets from node A to node C is 51, the minimal queue length of packets from node A to node C is 14.

The experiment proves that the congestion control algorithm responds quickly, when congestion occurs in a mid-node, the node will request previous node to reduce packet sending rate. It is faster than traditional packet drop—overtime retransmission—reducing packet sending rate strategy. The algorithm reduces packet drop in transmission effectively. In active network, mid nodes may have spent much time on computing a packet, It cost much to discard the packet. We always try to avoid discarding packets in our congestion control strategy. Unlike traditional end-to-end congestion algorithm, our Hop-to-hop Feedback congestion algorithm let more mid-nodes take part in congestion control. Simulation results show that no matter what the original status is, the network will reach dynamical balance in short time. The computation resources of a node are utilized sufficiently, and computation resources can be allocated dynamically.

The control granularity of current model is still rough, and is based on nodes, but not data flow, in order to achieve more precise control, we will construct more complex model in the next step.

## 5. Conclusion

Traditional end-to-end congestion control can't meet the need of active networks. According to characters of active networks, the Congestion Control Model Based on Hop-to-hop Feedback is provided for active networks. The working process is a node detects congestion by monitoring the queue buffers, then sends control packet to neighbor nodes, the correlative nodes will cooperate to control congestion. Theoretic analysis and simulation experiment express that the solution can eliminate congestion quickly, and can reduce packet drop effectively.

Essential difference between traditional networks and active networks is the mid nodes of the latter have the ability of computation, we should pay more attention to inner information of networks. Our research is still in primary stage, in the next step, we should change control granularity from node to data flow. That will relate to perfect control model and optimize control strategy. Future networks will enhance mid nodes' ability of computation, our research may provide reference to congestion control for future networks.

## References

- [1] Tennenhouse D L, Smith J M, Sincokie W D, et al. A survey of active network research[J]. IEEE Comm. Mag., 1997,35(1):80~85.
- [2] Jacobson V. Congestion avoidance and control. ACM Computer Communication Review, 1988,18 (4):314-329
- [3] Theodore Faber. Experience with Active Congestion Control[A]. DANCE 2002[C]. Univ. of Southern California: 2002,132-142.
- [4] Tibor Gyires. Using active networking for congestion control in high-speed networks with self-similar traffic[A]. Systems, Man, and Cybernetics, 2000 IEEE International Conference on[C]. Nashville, TN, USA, 2000,405-410
- [5]Rima Kilany ,Contributions / J-Sim Add-Ons[EB/OL]. <http://www.j-sim.org/contribute.html#anet>. August 9,2003.



databases.

**Nie Fei** received the B.S. and M.S. degrees in Electrical Engineering from Xi'an Engineering College in 1992 and 1996, respectively. Currently, he is a Ph.D. student in Institute of Computer Architecture and Networks at Xi'an Jiao Tong University. His research activities include computer networks, distributed