ISSN:0975-9646

Improved Hop-by-Hop Congestion Control Algorithm

Urvashi Handa#1, Padma Bonde#2

[1]*Computer Technology and Application,*
Chhattisgarh Swami Vivekanand Technical University, India
[2]*Computer Technology and Application,*
Chhattisgarh Swami Vivekanand Technical University, India

Abstract— **Network congestion plays an important role in degrading the performance of the communication network. Thus it is important to detect and control congestion in order to improve the performance of the network. There are various sources for congestion like packet collision, buffer overflow, concurrent transmission etc. This paper focuses on hop by hop congestion control mechanism. The proposed protocol is very efficient in detecting and controlling congestion. Every time a new packet arrives at the router, it keeps track of the number of packets in the queue. Based on the measured value the router effectively adapts its transmission data rate to control congestion.**

Keywords— **Hop-by-hop congestion control mechanism, congestion detection, congestion control**

I. Introduction

In the growing world of networking, more and more emphasis is being placed on speed, connectivity, and reliability. Networking has influenced our everyday lives, as the interconnectivity of families and friends has changed the ways in which they communicate and seek information. With the growing number of network users and concurrent transmission of packets, chances of network congestion have increased. Network Congestion, which is quite common in communication networks, occurs when demand for network resources exceed the available capacity. It occurs when a link or node is carrying so much data that its quality of service deteriorates. It leads to loss of packets at buffer, require retransmission, increased delays and energy wastage. This results in degradation of the performance and lifetime of network.

To overcome the problem of network congestion, congestion avoidance and congestion control techniques are needed to ensure network availability and efficiency. Modern networks use congestion control and congestion avoidance techniques to try to avoid congestion collapse. The reason behind congestion detection and control is to provide high data rate transmission with high efficiency and reliability. When the network has less traffic, network is not prone to congestion whereas when the traffic increases, network is prone to congestion degrading the overall efficiency of the network. This increases the need to develop a protocol to detect and control congestion. Congestion occurs when huge number of sensed data flows through same path to reach the destination node or sink node, thus leading to bottleneck.

There are three ways to control congestion: Proactive control, Reactive control and Hybrid scheme. Proactive congestion control tries to prevent congestion. In proactive congestion control, the congestion control mechanisms make reservations of network resources i.e. bandwidth and buffers. Network can request that each user specify a performance requirement, and can reserve resources so that this level of performance is always available to the user. Therefore, proactive congestion control is also known as Reservation-Oriented congestion control. Reactive control detects the congestion and then takes necessary actions to overcome the problem of congestion. In this the users are allowed to send data without reserving resources, but with the possibility that, if the network is heavily loaded, this may lead to congestion. In a reactive congestion control, the owners of conversations need to monitor and react to changes in network state to overcome congestion. Hybrid schemes can combine aspects of both approaches. One such hybrid scheme is for the network to provide statistical guarantees. For example, a user could be guaranteed an end to end delay of less than 10 seconds with 0.9 probability. Such statistical guarantees allow a network administrator to overbook resources in a controlled manner. Thus, statistical multiplexing gains are achieved, but without completely giving up performance guarantees. Another hybrid scheme is for the network to support two types of users: guaranteed service users and best-effort users. Guaranteed service (GS) users are given a guarantee of quality of service, and resources are reserved for them. Best-effort (BE) users are not given guarantees and they use up whatever resources are left unutilized by Guaranteed service (GS) users.

As the growth rate of internet and its traffic continue, the issue of congestion detection and reduction becomes increasingly important. The ideal goal is to be able to reliably detect all congested nodes or links in the network. There exist many congestion control techniques but search for new techniques continue. Two congestion control techniques, Choke packets and Hop-by-hop congestion control mechanism are discussed here. Choke packets is Reactive congestion control mechanism. The basic idea of choke packet is that the router checks the status of each output line. If it is too occupied, router sends a choke packet to the source. The host is assumed to be cooperative and will slow down. When the source gets a chock packet, it cuts rate by half, and ignores further choke packets coming from the same destination for a fixed period. After that period has expired, the host listens for more choke packets. If one arrives, the host cut rate by half again. If no choke packet arrives, the host may increase rate. The problem of basic Choke Packets are: i) Generating a packet is a difficult for a router, while it is overloaded. ii) For high-speed WANs, return path for a choke packet may be so long that too many packets have already been sent by the source before the source notes congestion and takes action.

In Hop-by-hop congestion control (sometimes also called *backpressure*), each router (hop) along an end-to-end path sends feedback to the directly preceding router, which executes control mechanisms based on this feedback. In general, the goal of hop-by-hop schemes was to attain reliability *inside* the network, for example, by buffering packets at each hop until they can be sent on (because there is no more congestion) and only transmitting as much as the next hop allows.

Applications requiring high data-rate can easily cause congestion problem especially at intermediary nodes. The proposed protocol detects the congestion at intermediary nodes and controls in order to provide both huge data transfer and high reliability.

II. RELATED WORKS

In [1], congestion is detected by measuring the queue length. The congestion is controlled by using three techniques i) hop-by-hop flow control , ii) source rate limiting, and iii) prioritized MAC. Even in high offered load it claims to achieve good throughput and fairness. In [2], a congestion control technique in which packet service time is used to infer the available service rate and therefore detects congestion in each intermediate sensor node. The congestion is controlled by hop-by-hop technique and it uses rate adjustment based on the available service rate and number of child nodes. However, it cannot utilize the available link capacity efficiently when some nodes are in sleep state. An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it.

In [3] the author proposed a metrics called Depth of Congestion (DC) to detect congestion. The Depth of Congestion (DC) is given as input to the congestion detection unit. When the measured DC is less than pre-defined threshold value, then, there is no congestion whereas when DC is equal to or more than pre-defined threshold value, then it intimates the occurrence of congestion. Once congestion is detected, notification signal is send to all the nodes to control the congestion. On receiving the notification signal each node adjust the transmission rate by implementing Hop-by-Hop Rate Control Technique.

The methodology in [4] has given a efficient method for both congestion detection and avoidance, the authors gets the notification of congestion by taking a term Intelligent Congestion Detection (ICD), the packet service time and inter arrival time is taken into account for the detection. When the above method ends in the congestion phase a notification to all the nearby nodes is been provided by the method called implicit congestion notification (ICN). The avoidance of congestion is been made out by assigning (PRA) priority by taking the packet delivery rate as key.

In [5] Present a congestion avoidance protocol, which includes *source count* based hierarchical medium access control (HMAC) and weighted round robin forwarding (WRRF) as the two main method to overcome congestion . The Simulation result of the this technique avoid packet drop due to buffer overflow and achieves much higher delivery ratio even under high traffic condition, which claims a good enough method for reliable event detection.

III. SYSTEM OVERVIEW

The main motivation in designing this protocol is to present an effective and efficient mechanism to detect and control congestion. To achieve reliability and efficiency in networking a mechanism that is proposed is Improved Hop-by-hop congestion control algorithm.

In the proposed protocol, each internet router maintains a queue. Queue length depends on bandwidth (i.e. link capacity), delay (i.e. average RTT of flows) and number of flows in the network. Every time a new packet arrives at the router, it keeps track of the number of packets in the queue. If the number of packets in queue is beyond a parameter "min", then the router sends a hop-by-hop choke packet to the source, requesting to reduce rate. If the number of packets is beyond a parameter "max", then router will drop a packet after every specific interval of time

A. Algorithm Description

```
 Assume Queue as Circular Queue
F and R are the front and rear in circular queue respectively
Flag is a variable for congestion notification
Set Queue length = Q
Set min = └ (2/3 of Q) ┘
     max = ┌ (4/5 of Q) ┐
While
          Packets are being transmitted in the network &
received at the routers.

do
          Set Flag=0;
          [Calculate no. of packets]

          if (F<R)
                    N= [(Q-F)+R+1]
          else
                    N=R-F+1

          [Compare No. of packets with ' min' & ' max']

          if (N ≥ min & N<max)
          then
                    [Packet arrival rate is greater than packet
                    service rate]
                    [Congestion may occur]
```

A packet will be send to sources to reduce transmission rate.

Set Flag=1 [1 indicates Congestion]

elseif (N ≥ max)
then

[Packet arrival rate is much greater than packet service rate]
[Congestion will definitely occur in sometime]

Router will drop 1 packet after every specific interval of time.

else

[Packet arrival rate is smaller than packet service rate]

No congestion

End if
End while

B. Example

If the calculated number of packets in the queue is between parameter 'min' and 'max' then congestion may occur, therefore Flag will be set to 1 indicating congestion. If the calculated number of packets in the queue is beyond parameter 'max' then congestion may definitely occur so the router will drop 1 packet indicating the source to reduce transmission rate. If the calculated number of packets in the queue is below parameter 'min' then no congestion has occurred source may continue sending packets at same rate.

Consider a router having a queue with length 30. So by the mentioned formula parameters 'min' and 'max' will be calculated as 20 and 24 respectively.

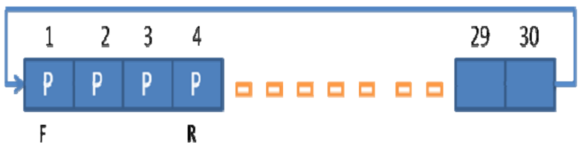

Fig. 1 No Congestion

Here the number of packets, N is found to be 4, which is less than 'min', therefore there is no congestion.

Fig. 2 Number of packets in queue is greater than 'min'

In above figure, number of packets, N is calculated to be 21, which is between the values of min and max, so router will set Flag=1.

Fig. 3 No of packets greater than 'max'

In above figure, number of packets, N is calculated to be 28, which is greater than max, so router will drop 1 packet after every specific interval of time.

IV. Conclusion and Future Work

Congestion control is a very important issue in computer networks. The proposed work helps in predicting congestion and improving efficiency and reliability in the network. It helps in reducing packet loss, reduced transmission delay and according to the protocol source will reduce its rate earlier and the queue will not overflow.

Future work will focus on finding the most congested flows and stop only those flows without affecting the non congested flows. By this we can achieve greater reliability and efficiency in the network.

References

[1] B. Hull, K. Jamieson, and Balakrishnan, "H. Mitigating Congestion in Wireless Sensor Networks", In Proceedings ofACM SenSys, pp. 134-147, 3-5 November 2004.

[2] C.T. Ee and R. Bajcsy, "Congestion Control and Fairness for Many-to-One Routing in Sensor Networks", In Proceedings of the ACM SenSys, pp. 148-161, 3-5 November 2004.

[3] R. Chakravarthi and C. Gomathy "*Hop-by-Hop Rate Control Technique for Congestion Due to Concurrent Transmission in Wireless Sensor Network",* World of Computer Science and Information Technology Journal(WCSIT), ISSN: 2221-0741, Vol. 1, No. 8, pp. 351-356, 2011.

[4] C.Y. Wan, S. B. Eisenman, A. T. Campbell, and J. Crowcroft, "Overload Traffic Management for Sensor Networks", ACM Transactions on Sensor Networks, Vol. 3, No. 4, Article 18, October 2007.

[5] Y. Sankarasubramaniam, A. Ozgur, I. Akyildiz, "ESRT Event-to-Sink Reliable Transport in wireless sensor networks", In the proceedings of ACM Mobihoc, pp. 177–189. ACM Press, New York (2003)

[6] M. Zawodniok and S. Jagannathan "Predictive Congestion Control Protocol for Wireless Sensor Networks", IEEE Transactions on Wireless Communications, Vol. 6, No. 11, pp. 3955-3963, 2007.

[7] M. Welzl, *Network congestion control: managing Internet traffic,* John Wiley & Sons Ltd, 1973.

[8] K. Sharma, Dr. H. Singh and Dr. R.B. Patel, "A Hop by Hop Congestion Control Protocol to Mitigate Traffic Contention in Wireless Sensor Networks", International Journal of Computer Theory and Engineering, Vol. 2, No. 6, pp. 986-991, December 2010.

[9] Y. Yi and S. Shakkottai, "Hop-by-hop Congestion Control over a Wireless Multi-hop Network", Proc. IEEE Infocom, Vol. 4, pp. 2548-2558, March 2004.

[10] K. Tan, F. Jiang, Q. Zhang and X. Shen, "Congestion Control in Multihop Wireless Networks", IEEE Transaction on Wireless Communications, Vol. 56, Issue. 2, pp. 863 – 873, 2007.

[11] Y. Hsu and K. Feng, "Cross-Layer Routing for Congestion Control in Wireless Sensor Networks", Radio and Wireless Symposium, pp. 783-786.

[12] C. Wang, Y. Eisenman and A. Campbell, "CODA: Congestion Detection and Avoidance in Sensor Network, Proc. of ACM SenSys, pp. 1003-1016.

[13] J. Wan, X. Xu, R. Feng and Y. Wu, "Cross Layer Active Predictive Congestion Control Protocol for Wireless Sensor Networks", Sensors 2009, Vol. 9, No. 10, pp. 8278-8310, 2009.

[14] Y. Yi and S. Shakkottai, "Hop-by-hop Congestion Control over a Wireless Multi-hop Network", Proc. IEEE Infocom, Vol. 4, pp. 2548-2558, March 2004.
[15] R. Jain, "Congestion Control in Computer Networks: Issues and Trends", IEEE Network Magzine, pp. 24-30, 1990.
[16] R. Banner and A. Orda, "Multipath Routing Algorithm for Congestion Minimization". IEEE/ACM Transaction on Networking, Vol. 15, No. 2, pp. 413-424, 2007.
[17] ELEC3030 (EL336) Computer Networks S Chen, Available:users.ecs.soton.ac.uk/sqc/EL336/CNL-11.pdf.
[18] Congestion Control in Computer Networks – EECS Technical Reports, Available:techreports.lib.berkeley.edu/accessPages/CSD-91-649.html.