International Journal of Engineering Research & Management Technology

May- 2015 Volume 2, Issue-3
ISSN: 2348-4039

Email: editor@ijermt.org www.ijermt.org

"Sharpening Skills..... Serving Nation"

Special Issue: 2nd International Conference on Advanced Developments in Engineering and Technology
Held at Lord Krishna College of Engineering Ghaziabad, India

Analysis the Performance of Drop Tail-AQM Mechanism Techniques

Anmol Kumar
M.Tech Scholar
Al- Falah School of Engg. & Tech.
Faridabad, Haryana

Sher Jung Khan
Ph.D. Scholar
Al- Falah School of Engg. & Tech

Nidhi Jaiman
B.Tech.Scholar
Computer Science and Engg.
Biyani International Institute of Engg. & Tech
Jaipur (Rajasthan)

ABSTRACT—
Queue Management is a way to control Congestion. The design of Queue Management Scheme for routers in the internet has following factors: queuing delay, link utilization, packet loss and the impact of the router buffer size. Drop Tail Mechanism Techniques used with TCP Protocol that are commonly used in now a day's network. The implementation of Drop Tail is easy but has the problems of lockout and Synchronization.

Keywords— Drop Tail; Queue management; Congestion control; Congestion avoidance; Throughput

I. INTRODUCTION

Congestion is a problem that happens in networking when there is so much data that the network cannot burden anymore. It has a huge influence to both wired network and wireless network and causes the problems of packet loss, packet delay and lockout. A long time could be taken to recover from that situation.

To control congestion, several techniques are used, such as exponential back off, congestion control in TCP, priority schemes, and queue management. Exponential back off is used in CSMA/CA, which is the sensing scheme of 802.11. The sender senses the channel before transmission. If the channel is busy, it waits until idle and sends the data after a random period. The random period is calculated by exponential back off.

Congestion control in TCP consists of slow start, congestion avoidance, fast retransmit and fast recovery [7]. It is a method to control the transmitting rate of the sender. The TCP flow starts at a very low rate and increases exponentially to a threshold. Congestion avoidance then happens and the congestion window increases by one segment each time for one successful transmission. Fast retransmit is first used in TCP Tahoe to retransmit lost packet. And fast recovery is first appeared in TCP Reno after the step of fast retransmit to skip the slow start.

Priority scheme marks packets into different priorities and drops low priority packets when it is needed. It is not a real congestion control method but improves the performance with other methods.

Queue management is a way to control the queue size of the bottlenecks. It contains passive queue management, which drops packet when the queue is full; and active queue management, which drops packet before the buffer getting full. Drop Tail is an algorithm that represents the way.

Drop Tail mechanism is used by routers that when packets should to be drop. In this mechanism each packet is treated identically and when queue filled to its maximum capacity the newly incoming packets are dropped until queue have sufficient space to accept incoming traffic. When a queue is filled router start to discard all extra packets thus dropping the tail of mechanism. The loss of packets causes the sender to enter slow start which to decrease the throughput and thus increases its congestion window.

Queue management is used to control and optimize queues. In networking, it is needed when several nodes transmit data to a bottleneck link. The following figure is a basic topology of network. The bandwidth between sources and Router 1 is 10 Mb, but the bandwidth between routers is 2 Mb. So the link between routers is the bottleneck link. When packets from Source 1 and Source 2 arrive to Router1, they queue and wait for transmission. However, the buffer size is limited. If the buffer is full and the sources keep transmitting, congestion will occur. It will cause congestion collapse and lockout, and the network will restore after a long time [1].

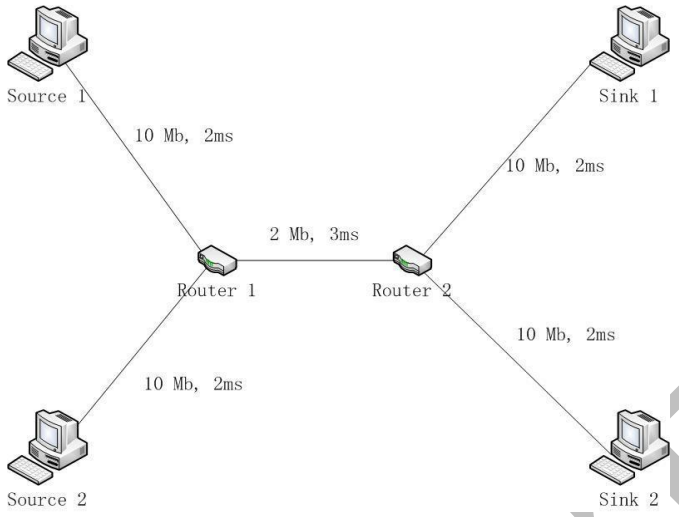


Figure1. Basic topology of network

Queue management is one of the methods to control congestion. It controls the queue length of the buffer when certain condition is reached. Besides that, CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) with exponential backoff and TCP congestion control are used in wireless networks.

Active vs. passive queue management algorithms: The buffer size cannot be too large to increase the packet delay, but we can control the queue size by queue management algorithms. In this thesis, we only consider the algorithms related to TCP.

There are two basic methods of queue management [25]. One is called passive queue management. In this method, packets are dropped only if the buffer is full. It contains several algorithms, such as Drop tail, Head drop, and Push out [2] [3]. Drop Tail is the most common algorithm of passive queue management. It drops packet from the tail of the buffer when the queue is full, and does nothing when the buffer still has space. Head drop drops packets from the front of the queue. The delay time is less than that of Drop Tail because it drops old packets and keeps new packets. Push out is an algorithm that pushes the last queued packet out and puts the coming packet in if the buffer is full.

Passive queue management algorithms are easy to implement in real networks. However, some disadvantages need to be mentioned. First of all, the average queue length of passive queue management algorithms is large for a long period time. As a result of that, the end-to-end delay is long. Second, passive queue management is a way of congestion control, but not a way of congestion avoidance. The sources reduce the rate of transmitting when the queue is full. Those packets that transmitted before the reduction are all dropped and need to be retransmitted. Moreover, passive queue management algorithms cause the problem of lock out. In this case, a single connection transmits normally while others decrease their rates. Then the buffer is full of the packets from that connection. The fairness cannot be guaranteed. To avoid congestion and lock out, active queue management algorithms are proposed. Those methods begin to control the queue length before the buffer getting full. The algorithm that widely used is RED [4] [5]. In this algorithm, packet is dropped with a given probability when the average queue size achieves the threshold; and the probability increases when the average queue length becomes long. By doing that, the sender cannot get the acknowledgement of the dropped

packet from the receiver, and will reduce the transmission rate and adjust the cwnd of TCP protocol before the queue getting full.

II. CSMA/CA AND EXPONENTIAL BACKOFF

Collisions may happen at the base station if two messages arrive in the same time slot. The more the amount of nodes is, the more the number of collisions is. In 802.11 networks, CSMA/CA is used to prevent collisions. The processes are showed as figure 2.

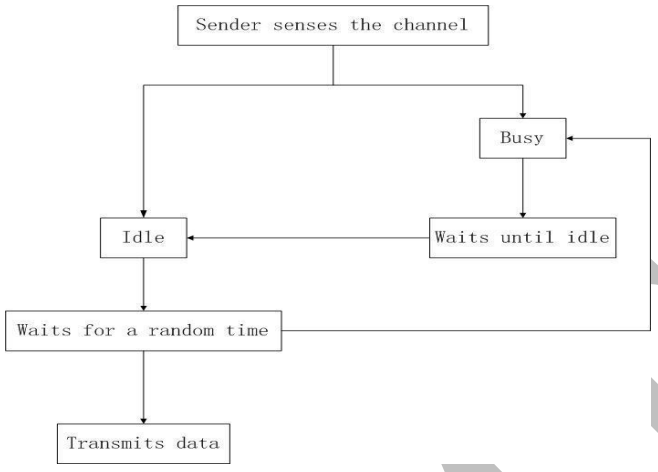


Figure2. Process of CSMA/CA

Before transmitting, the sender senses the channel and determines whether the medium is idle or not. If the channel is idle, the sender counts a randomly time using exponential back off and then transmits the data. Otherwise it waits until the channel becomes idle, and sends the data after a randomly chosen duration counted by exponential back off, before attempting the transmitting again [9] [20] [21]. Although CSMA/CA reduces the probability of collisions, they still happen during simulation.

RTS/CTS handshake is another technique used in CSMA/CA. before transmitting data; the sender should transmit RTS (Request to send) to the receiver. And the receiver sends CTS (Clear to send) back to the sender. The overloads of RTS and CTS packets are small. This reduces the overload of invalid transmission. RTS/CTS also avoid the problem of hidden nodes. The process of handshake can be observed in the trace files of NS2.

In CSMA/CA, exponential back off is used to generate the random duration. This is to protect the channel from a bursty traffic when several senders sense the idle channel and transmit together. Thus, exponential back off reduces the probability of collision when there is a large amount of senders. Exponential back off delay is an integer multiple of time slot. The number of slots to delay depends on a uniformly distributed random integer r as [10] [22]:

III. PROBABILITY OF DROPPING IN DROP TAIL MECHANISM

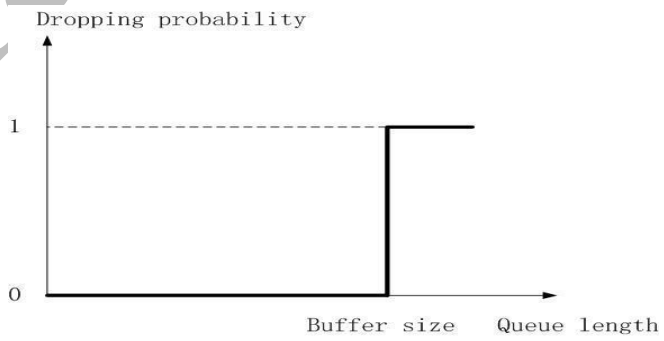


Figure 3. Dropping probability of Drop tail

Drop tail is the common algorithm of passive queue management. It drops all the new packets when the buffer is full, and does nothing when the buffer still has space [3] [6]. The figure above shows the dropping probability of packets. The only two dropping probabilities are 0 and 1. When the number of packets arrived to the queue larger than the buffer size, the probability of packet dropping is 1. Otherwise the dropping Probability is 0. The algorithm is easy to implement and does not have complicated parameters. But it also gets the problems of lock out and synchronization. In lock out situation, most of the connections decrease their transmitting rates except one or few of them. Then the buffer is full of packets from connections with high rates. Synchronization is different. Nearly all connections increase or decrease their transmission rates together. The packets will have a huge delay and loss in this situation. The average queue length is always high if there are a large number of sources.

IV. TCP CONGESTION CONTROL

As we concentrate on the queue management algorithms with TCP protocol, we should understand TCP congestion control as well. Besides the control of queue, TCP protocol starts to control congestion from transmitting rate. The processes are: slow start, congestion avoidance, fast retransmit and fast Recovery [8] [15] [23] [24].

a. Slow start: The TCP protocol starts with a very low rate to ensure the success of transmission. Then the rate grows exponentially to reach the slow start threshold (ssthresh). The process ensures that the bandwidth is fully utilized.

b. Congestion avoidance: After reach the ssthresh, the rate increases linearly, i.e. one full-sized segment each time, until one packet lost. The process avoids bursty traffic and large amount of packet loss.

c. Fast retransmit: Receiver sends duplicate ACKs with the same segment number if one packet has lost and the sender keeps transmitting. If the sender receives three duplicate ACKs, it will retransmit the packet without waiting for the RTT. Fast retransmit is applied in both TCP Tahoe and TCP Reno.

d. Fast recovery:

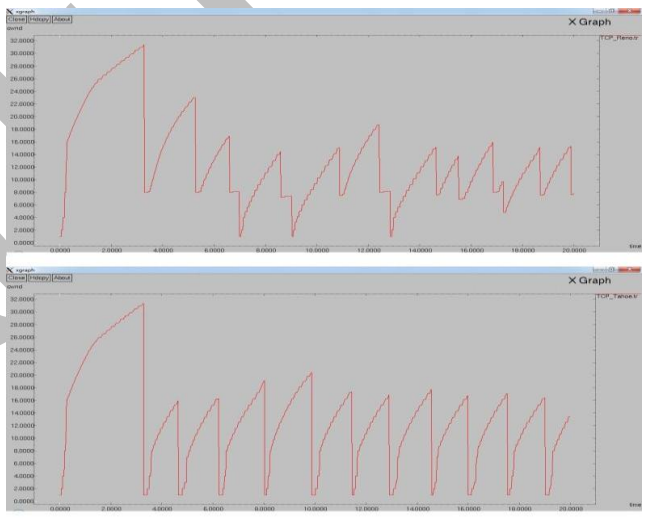


Figure4. Comparison of TCP Reno and TCP Tahoe

Fast recovery is first used in TCP Reno. Unlike the ssthresh of TCP Tahoe, which restarts from one every time, the ssthreshof TCP Reno restarts from: max (Flightsize/2, 2*MSS) as showed in figure 2.4. And the congestion window (cwnd) equals to ssthresh+3. It is obvious that TCP Reno gains a better performance than TCP Tahoe.

V. THROUGHPUT OF DROP TAIL

The throughput of the receiver is obtained by monitoring the TCP sink [30]. The expression of throughput in Mbps is:

$T=B\ (8/t*10^6)$ ---1.1

Where T is the throughput, B is the amount of bytes that has been received, and t is the simulation time.
In this simulation, we record the average throughput between Sender 1 and Receiver 1 and write it in a trace file. The interval of each sample is 0.2 s. the layout of the trace file is:

60.000000000000313 0.27120533333333191
60.200000000000315 0.272497009966776
60.400000000000318 0.27278675496688598
60.600000000000321 0.2742627062706256
60.800000000000324 0.27355789473684067

The first row is the simulation time and the second row is the average throughput from the beginning to that moment.

Figure 5. Throughput of Drop tail with 5 nodes and 0.2 packet loss prob The above figure presents the throughput of Drop tail and RED. Those graphs are printed by analyzing the trace file. The green lines are instant throughput and the red lines are average throughput. With the graphs, we can clearly see how the throughput changes. The data of throughput is statistically counted.

VI. AVERAGE QUEUE SIZE

The average queue size of the buffer is got by an awk code which analyses the trace file line by line. The information in a trace file is as following [18]:

+ 0.020288 0 3 tcp 1500 ------- 1 0.0 5.0 1 6
- 0.020288 0 3 tcp 1500 ------- 1 0.0 5.0 1 6
+ 0.020288 0 3 tcp 1500 ------- 1 0.0 5.0 2 7

The information we need is in row 1, 2, 3, 4, 6, and 12:
a. Action: The first row is action row that indicates if the packet is arrived, departed, received or dropped. It has four symbols: '+' (arrived), '-' (departed), 'r' (received), and 'd' (dropped).
b. Time: The second row presents the time that the action happens.
c. Source node: The node that transmits that data is showed in row 3. '10' is the node address of the source node. It has to be clear that the source node is not always the real sender of that data. It could be the router or any forwarder on the link.

d.Destination node: The source node is followed by the destination node in row 4. Like the source node, this is not always the real receiver of the data. The destination node could be any node that forwards this data to the receiver.

e. Packet size: The sixth row is the row of packet size. The default packet size of TCP in NS2 is 1000 bytes. But the common packet size of real network is 1500 bytes including a 40 bytes header.

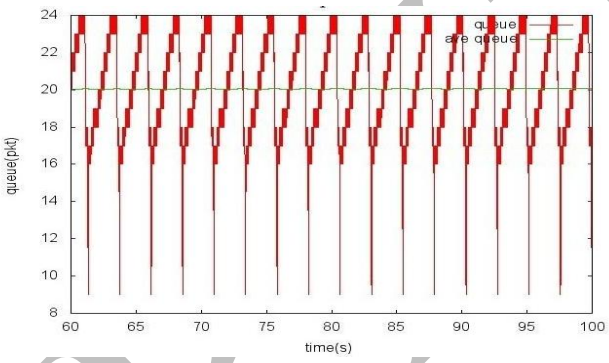
f. Packet ID: The last row of the trace file is the row of packet ID. Each packet has its own packet ID in the simulation.

The awk code analyses each line of the trace file and counts the packet sent, received and dropped respectively by judging the action, the source node and the destination node. TCP packets are separated from ACKs by the packet size. And the

Packet lost is indicated by packet ID. The expression of the average queue length is:

$q=a-d-1$ --- 1.2

Where **q** is the average queue length, **a** is the amount of arrived packet, **d** is the amount of departure packet, and l is the amount of packet dropped. The simulation time is 100 seconds. However, each simulation model has the period getting stable. The period is called warm up period. To avoid the time of warming up, the data we utilized is from 60 seconds to 100 seconds.



VII. CONCLUSION

In this paper, to analysis the performance of the drop tail mechanism. So drop tail has a small queue size that represents the short delay. Due to this problem is created when a queue is filled then the router is discarded all extra packets and due to the causes of the loss of the packets the sender to enter slow start which decreases the throughput and increase the congestion window. To solve this problem comes Random Early Detection (RED) Mechanism. It is another type of Active Queue Management technique that overcomes the problem of drop tail AQM mechanism. It is also a congestion avoidance queuing mechanism that is potentially useful, particularly in high-speed transit networks. It operates on average queue size and drop packet on the basis of statistics information.

APPENDIX

ACK: A flag used in TCP to acknowledge receipt of a packet
AGT: Agent trace in trace file
AQM: Active Queue Management
CSMA/CA: Carrier Sense Multiple Access with Collision Avoidance
Cwnd: Congestion window
Drop tail: Drop from tail
Head drop: Drop from head
NS2: Network Simulator 2
RTR: Router trace in trace file
Ssthresh: Slow start threshold
TCP: Transmission Control Protocol
WIFI: Wireless Fidelity

REFERENCES

1. John Nagle. RFC 896 – Congestion Control in IP/TCP Internetworks. Tools.ietf.org.1984-01-06. http://tools.ietf.org/html/rfc896
2. Mujdat Soyturk. Design and Analysis of TCP/IP Networks. Gebze Institute of Technology. Spring 2011. Available at: http://www.gyte.edu.tr/hebe/AblDrive/99104018/w/Storage/104_2010_2_574_99104018/Downloads/week3-4activequeuemanagement.pdf
3. Dmitri Moltchanov. TLT-2727: Queue Management. Tampere University of Technology
4. Sally Floyd, Van Jacobson. Random Early Detection Gateways for Congestion Avoidance. IEEE/ ACM Transaction on Networking. August 1993
5. Mats Sagfors, Reiner Ludwig, Michael Meyer, Janne Peisa. Queue Management for TCP Traffic over 3G Links. Wireless Communications and Networking, 2003.
6. Xinnian Chen. Analysis and Comparison of the NS2 Based Router Algorithms: Droptail. Computer Engineering & Science, vol. 29, No. 6, 2007
7. Roman Dunaytsev. Congestion Control. Tampere University of Technology. 2010-12-02
8. Roman Dunaytsev. MAC Techniques. Tampere University of Technology. 2010-10-07
9. M. Allman, V. Paxson, W. Stevens. RFC 2581 – TCP Congestion Control.Tools.ietf.org. April 1999.
10. IEEE Standard 802.3-2008". Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications. NY 2008. IEEE-SA Standards Board. 67 P. http://standards.ieee.org/getieee802/download/802.3-2008_section1.pdf
11. Jae Chung, Mark Claypool. NS by Example. Worcester Polytechnic Institute. http://nile.wpi.edu/NS/
12. Marc Greis. Tutorial for the Network Simulator "ns". http://www.isi.edu/nsnam/ns/tutorial/index.html
13. Kevin Fall, Kannan Varadhan. The ns Manual. UC Berkeley, LBL, USC/ ISI, Xerox PARC. 2011-11-04. 126 P. http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf
14. NS-2 Tutorial. Demokritos University o Thrace.49 http://skontog.gr/work/wlesson.pdf
15. Arijit Ganguly, Pasi Lassila. A Study of TCP-RED Congestion Control Using NS2. 2001-07-20
16. General main RED parameters. http://www.cs.technion.ac.il/Courses/Computer-Networks-Lab/projects/spring2003/ns1/Net%20Site/Source/RedParam.htm
17. Sally Floyd. RED: Discussions of Setting Parameters. Nov 1997. http://www.icir.org/floyd/REDparameters.txt
18. NS-2 Trace Formats. National Science Foundation. http://nsnam.isi.edu/nsnam/index.php/NS-2_Trace_Formats
19. Teerawat Issariyakul, Ekram Hossain. Introduction to Network Simulator NS2.Springer. 2008-10-20. 332 P.
20. Pablo Brenner. A Technical Tutorial on the IEEE 802.11 Protocol. BreezeCOM. 1997. http://www.sss-mag.com/pdf/802_11tut.pdf
21. 802.11 MAC (Media Access Control). ZyTrax. http://www.zytrax.com/tech/wireless/802_mac.htm
22. CSMA/CA. http://www.cs.clemson.edu/~westall/851/802.11/802_CSMA_CA.pdf
23. TCP Congestion Control. http://condor.depaul.edu/jkristof/technotes/congestion.pdf