PERFORMANCE EVALUATION ON DROP-TAIL AND RED GATEWAYS

Piyatamrong B., Anutchalakhom W.

Department of Computer Engineering, Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang
Chalongkrung Rd., Ladkrabang, Bangkok 10520, Thailand
Phone: (662) 326-9969 Ext. 107, E-mail: kpbanjon@kmitl.ac.th, warut@octl.com

Abstract

The Drop-Tail and RED are well-known queue types deploy on gateways nowadays. RED is a newer queue type with the active queue management feature. They both manage to pass or drop the packet on different ways. A question on this 2 types is what case we choose to deploy which one. This paper uses simulations to explore the behaviors of Drop-Tail and RED gateways on various kind of traffic in term of throughput and packet dropping. We include both Reno TCP and SACK TCP, the last two standard implementations for TCP, in our simulations. The results show Drop-Tail performs better than RED on most case in our environments.

1. Introduction

In the world of Internet, TCP and IP are 2 key protocols. IP is used as a main protocol in network layer. TCP positions in transport layer that on top of IP with features of flow control, reliability and connection-oriented mechanism [2]. To improve the performance of TCP, the upper application layer protocols will receive the benefits also. The window-based end-to-end mechanism that used to maintain throughput in the network does not need cooperation from a gateway that just passes a packet across the network interfaces. TCP itself probes the network state to determine a size of congestion window [3] to utilize the network link bandwidth.

In [1, 4], Explicit Congestion Notification (ECN) has been proposed to appear on IP header and deploy onto IP gateways or IP routers for sending congestion notification comes along with IP header to notify congestion in the network. Firstly, ECN is used to enhance an active management queue, Random Early Detection (RED) [5, 6]. RED is still limited to deploy in the real world because of lack of information to setting 4 parameters (minimum threshold min_{th}, maximum threshold max_{th}, maximum drop probability max_p, queue weight w_q) to have optimal performance [7, 8].

2. The Explicit Congestion Notification

To support the ECN mechanism, totally 4 more bits on both IP and TCP header are redefined. The 2 bits on the IP header are ECN-Capable Transport (ECT) and Congestion Experienced (CE). The ECT provides a flag to notify that both the end systems are ECN capabled and the CE is set by a gateway when congestion occurs. The others 2 more bits on the TCP header are ECN Echo (ECE) and the Congestion Window Reduced (CWR) .The ECE bit is a flag that is bounced back to notify that the congestion is occurred and the CWR signal is used to confirm that the congestion window has been reduced. The order of the flows in the ECN mechanism is shown in Figure 1.

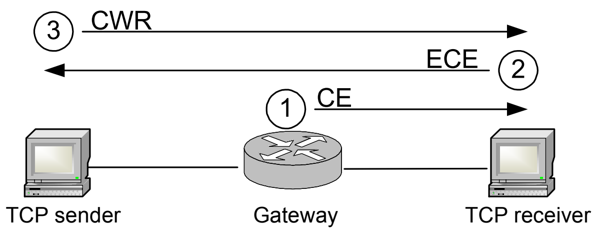

Fig. 1. Ordering flows of ECN mechanism.

The Drop-Tail with ECN (DT-ECN) is expressed in the term of algorithm as the following.

```
ECN_threshold = MAX_queue_size * (Queue_utilization / 100)
for each packet arrival
    if queue_length < MAX_queue_size
        if queue_length+1 >= ECN_threshold
            mark CE bit on the arriving packet
        enqueue the packet
    else
        drop the arriving packet
```

Where:

ECN_threshold:	level of queue that begins to mark CE bit.
MAX_queue_size:	maximum queue size.
Queue_utilization:	percentage of queue that begins to mark CE bit.
queue_length:	size of in used queue.

3. The Simulation Processes

The simulation process is done by using a wide area network (WAN) topology that consists of a group of sources on the left that send data via 2 routers to a group of destinations on the right, as shown in Figure 2. The environmental setting are:

- All network links have the same 45Mbps bandwidth (T3 standard).
- The propagation delay is set to 10 msec.
- The TCP senders are implemented for both TCP styles, i.e. TCP-Reno and TCP Selective Acknowledgment (SACK).
- All TCP connections called by FTP applications that always have data to send (bulk-data connection).
- The maximum segment size (MSS) of TCP is set 1460 bytes.
- The TCP receiver's buffer size is set to the maximum 16-bit window size TCP protocol field.
- The ssthresh variable of TCP is initialized to 65536 bytes.
- All 2 routers buffer sizes are set to twice of the bandwidth-delay product (BDP) of the network (675 kbytes).
- The ECN thresholds are set to 25%, 50% and 75% of the buffer size respectively.

The Network Simulator version 2.1b9a (NS-2) is an object-oriented, discrete event driven network that used in this paper. With the above conditions, a newly queue object module, DT-ECN, is implemented in C++ language to incorporate with the NS-2 framework and OTcl scripts are written to setting up the topology, the object values and the time that all the events occurred. After running the codes, the results are stored in the term of a trace file that contains information of time, position of a packet, destination of a packet, type of packet and any useful information. The trace file is then passed the filtering and calculating processes using Awk scripts to generating the final results.

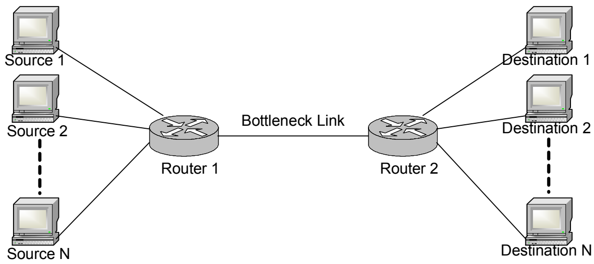

Fig. 2. Network topology for WAN in this simulation.

Various traffic patterns are used for the simulation. For instance:

- The total number of connections are varied from a very little to a very massive connections says 25 – 100 connections.
- The arrival and the departure of the number of connections is fluctuated from a little to a huge and falling down from a huge to a little number of connections and the final case is that all the connections enter at all time.

As defined in 6 different scenarios as A, B, C, D, E and F.

- Scenario A is a case that there are 10 connections at all the time of the simulation (0.0 seconds – 60.0 seconds) but in the middle of the period of time (20.0 seconds – 40.0 seconds) we increase traffic by 15 connections for 20 seconds.
- Scenario B is a case that the connections enter the network increasingly starting from 10 connections, 25 connections, and double to 50 connections.
- Scenario C is a case that inverses the scenario B by all the 50 connections enter the network aggressively at 0.0 seconds and decrease to 35 connections and 10 connections.
- Scenario D is a case that looks like scenario B with double of connection.
- Scenario E is a case that looks like scenario C with double of connection.
- The Last, Scenario F is a most aggressively case that all the 100 connections enter the network at all the 60.0 seconds period of time.

On the four RED parameters, the recommendation from Floyd in [5] as min_{th} = 1/4 of queue size, $max_{th} = 2 * min_{th}$, $max_p = 0.02$, $w_q = 0.002$ is employed.

4. Simulation Results

The measurement of the performance of the all 6 types of queue by summary of data that successful sent in byte (Bytes sent) and summary of data that has been dropped in byte (Bytes dropped). The results in term of value are shown in Table 1, 2, 3 and 4.

Table 1. Total Bytes sent on TCP Reno.

	Drop-Tail	DT-ECN @ 25%	DT-ECN @ 50%	DT-ECN @ 75%	RED w ECN	RED w/o ECN
	329588500	317098000	328577500	327248500	324961000	325607500
	329532500	315795500	328050500	328442000	323399000	322890500
	331103000	312348500	328025000	330624500	330620000	330560000
	330272500	313489000	329890000	329813500	326879500	324503500
	331523500	312826000	334180000	331286500	331148500	330953500
	332239000	312853000	333370000	331669000	331747000	331477000

Table 2. Total Bytes dropped on TCP Reno.

	Drop-Tail	DT-ECN @ 25%	DT-ECN @ 50%	DT-ECN @ 75%	RED w ECN	RED w/o ECN
	1695000	0	342000	681000	1122000	1972500
	2694000	0	342000	880500	2442000	3708000
	2680500	0	342000	681000	2160000	3636000
	5410500	0	409500	988500	6291040	7578040
	5607000	0	342000	681000	7126500	8013000
	9472500	0	342000	681000	12328500	13095000

Table 3. Total Bytes sent on TCP SACK.

	Drop-Tail	DT-ECN @ 25%	DT-ECN @ 50%	DT-ECN @ 75%	RED w ECN	RED w/o ECN
	335884000	317098000	336022000	335054500	335314000	333788500
	336149000	315795500	335399000	336561500	335573000	335276000
	336572000	312348500	332624000	336834500	334751000	334389500
	336428500	313489000	334150000	336919000	334721500	333767500
	336572500	312826000	333824500	336790000	335284000	336010000
	336985000	312853000	333311500	337174000	336437500	336433000

Table 4. Total Bytes dropped on TCP SACK.

	Drop-Tail	DT-ECN @ 25%	DT-ECN @ 50%	DT-ECN @ 75%	RED w ECN	RED w/o ECN
	1917000	0	342000	757500	1243500	2095500
	3351000	0	342000	991500	2302500	4021500
	2737500	0	342000	681000	2197500	3790500
	3729880	0	447000	859500	8332540	9576040
	5496000	0	342000	681000	9616500	10912500
	9306000	0	342000	681000	17812500	19065000

5. Conclusions

This paper evaluated Drop-Tail and RED gateways with bulk-data connection on WAN topology. On TCP, we used both Reno TCP and SACK TCP for covering most case of implemented TCP. We also added ECN onto both queue types for complete comparison. Firstly, Drop-Tail without ECN always performs better than RED without ECN in term of throughput and packet dropping. Secondly, Drop-Tail with ECN at 75% performs better than RED with ECN from 10 of 12 cases in term of throughput (Reno: A, B, C, D, E and SACK: B, C, D, E, F) and always performs better in term of packet dropping. The result is point out that Drop-Tail is still valuable to deploy at least on this environments.

References

[1] S. Floyd, “TCP and Explicit Congestion Notification,” *ACM Computer Communication Review*, Vol. 24 No. 5, October 1994, pp. 10-23.

[2] Information Sciences Institute University of Southern California, “TRANSMISSION CONTROL PROTOCOL,” *Request for Comments (RFC) 793*, September 1981.

[3] V. Jacobson, “Congestion Avoidance and Control,” *Proceedings of SIGCOMM 1988*, August 1988, pp. 314-329.

[4] K. Ramakrishnan, S. Floyd, and D. Black, “The Addition of Explicit Congestion Notification (ECN) to IP,” *Request for Comments (RFC) 3168*, September 2001.

[5] S. Floyd, and V. Jacobson, “Random Early Detection Gateways for Congestion Avoidance,” *IEEE/ACM Transactions on Networking*, August 1993.

[6] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang, “Recommendations on Queue Management and Congestion Avoidance in the Internet,” *Request for Comments (RFC) 2309*, April 1998.

[7] W. Feng, D. Kandlur, D. Saha, and K. Shin, “A Self-Configuring RED Gateway,” *INFOCOM '99*, March 1999.

[8] M. May, J. Bolot, C. Diot, and B. Lyles, “Reasons not to Deploy RED," *tech. rep.*, INRIA, France, June 1999.