

ECN Congestion Control Mechanism in IP Networks

Dr. Subarna Shakya¹, Anup Sainju²

¹Department of Electronics and Computer Engineering, Pulchowk Campus, IOE,
Tribhuvan University, Nepal

²Nepal Telecom

Corresponding e-mail: sshakya@wlink.com.np, anup.sainju@ntc.net.np

Abstract: Explicit Congestion Notification (ECN) is a newer method for congestion control in TCP IP networks. Network Simulator 2 (NS2) software has been used to compare the performance of ECN packet marking to other older and newer congestion control schemes, such as DropTail and Random Early Detection (RED), in both LAN and WAN schemes. During the simulations different parameters including proportion of packet drops, window size variation, queue size, and throughput were measured to evaluate the performance. The overall objective was to independently and comparatively study ECN in a wide range of situations to better understand its advantages and disadvantages. The results of these simulations showed that when all the network prerequisites were met (i.e. all the nodes including being ECN-aware), ECN reduced packet drops and thereby optimized network resource utilization and data throughput.

Key Words Explicit Congestion Notification, Network Simulator 2, Random Early Detect, DropTail

1. Introduction

1.1 Background

TCP is the predominant reliable data transfer protocol that drives the Internet and most other forms of computer networks today. It is reliable in the sense that it tends to ensure that every packet transmitted from source reach the destination. If a packet fails to reach the destination, the sender retransmits the packet. In a multi-hop, multi-node network such as the Internet, there may be different reasons why a packet from the source does not reach the destination. The main cause is network congestion (i.e. lack of sufficient bandwidth in the intervening links). Since a network with many nodes and links may have varying traffic load patterns, the source and destination of the data traffic should have adjustable rates at which they send and receive packets. When the network resources are free, they can transfer data in higher rates and when the resources are scarce they can decrease their rates. This flexibility of data rates at sending and receiving ends is facilitated by a TCP mechanism known as Windowing. Windowing means varying buffer size, especially at the sending end, so that the data sent out into the network fits the available resources there.

The general technique of congestion control is to prompt the sender to adjust its window size to avoid losses due to congestion. Internet congestion control has been referred to as one of the largest deployed artificial feedback mechanisms [1]. Conventionally, congestion control has been

a reactive measure, reducing the window size after detection of a loss or timeout. What ECN provides is a proactive approach to take preventive measures before any packet drops occur.

1.2 Research Methodology and Results

The tests and analysis performed are simulation based, using NS2 simulation software [2]. Numerous other software tools, such as Perl, AWK, and GNUPlot were used to analyze and graphically represent the simulation results in terms of tables, charts and graphs. The main objective of this research will be to analyze the performance of ECN as a proactive congestion control mechanism in TCP networks and document its effect on the overall network performance. The results will be used to recommend which scenarios ECN can and cannot be used as an effective tool for congestion control.

Simulations carried out with different network scenarios and parameters have shown that ECN in TCP networks is effective in reducing packet drops and hence improving efficient utilization of bandwidth. This result assumes that all intervening nodes between the sender and the receiver are also ECN capable, like the source and destination. However, if the intervening nodes are not ECN aware, a performance gain is not eminent. This is understandable given the way ECN and packet marking functions.

2. Congestion Control

2.1 TCP and Queue Management

Traffic management and congestion control mechanisms of TCP are host-specific by default (i.e. they are concerned with the source and destination of a data transfer, not the intervening network nodes such as routers). However, for actual minimization of congestion and efficient utilization of network resources, host-specific mechanisms need to be considered along with network-specific parameters. Concerning congestion, the most important parameter of a network device is the queue or buffer size, as well as the way it is managed.

The style of queue management directly impacts the way packets are dropped in the network. Hence, it is vital to look into queue management techniques while studying congestion control. There are two main variants of queue management techniques: DropTail and RED. ECN is a further refinement of RED with slight modifications in TCP and IP headers.

2.2 DropTail and RED

DropTail is a primitive form of queue management. As the name suggests, DropTail drops any packets that arrive when the buffer is full. This queue mechanism is usually deployed with TCP Reno which solely depends upon packet drop for congestion control.

Unlike DropTail, RED does not wait for a full buffer to start dropping packets. It maintains an exponentially weighted queue length and drops packets with a probability that increases with average queue length. When the average queue length is less than the minimum threshold no packets are dropped. When it exceeds the maximum threshold, all packets are dropped. When it is in between, a packet is dropped with a probability that is an increasing piecewise linear function of the average queue length. This type of strategy is called Active Queue Management (AQM) [3]. ECN can be called a further extension of RED, though queue management schemes other than RED can also be used with ECN.

3. ECN: Overview

The basic idea of ECN is to enable routers to send congestion-related information to data sending hosts so they can adjust their data rates to avoid congestion. Hence, ECN is a proactive congestion control mechanism designed to avoid congestion, as opposed to facing congestion and retransmitting lost or unacknowledged packets. Acting preemptively, ECN better utilizes network resources and improves network throughput while controlling congestion and packet losses.

For this mechanism to function, routers must be able to tell sending hosts that congestion is imminent so that senders can adjust their data rates to avoid or minimize congestion and resultant packet losses. ECN communicates this information in special data included in the network header of each packet. Specific guidelines for different formats of including this data in the packet header has been previously established [4][5][6]. Additional background information, various research notes, RFC, and more information about this relatively new approach to congestion control can be found in additional publications [7].

3.1 Addition of ECN to IP

For ECN integration, the concerned field is the ToS (Type of Service Field). Bit 6 & 7 of the ToS octet of the IP header are used for ECN [5][6]. The other 6 bits are already assigned for differentiated services fields [RFC 2474]. The bits 6 and 7 of ToS octet are used for to make 4 ECN codepoints [6].

Table 3.1 ECN Bits in IP Packet Header

ECN Bit 1 (ToS bit 6)	ECN Bit 2 (ToS bit 7)	ECN Codepoint
0	0	Not-ECT (non-ECN capable)
0	1	ECT (1)
1	0	ECT (0)
1	1	CE (Congestion Experienced)

The ECN-Capable Transport (ECT) codepoints '10' and '01' are set by the data sender to indicate that the end-points are ECN-capable. Routers treat ECT(0) and ECT(1) codepoints as equivalent. Senders are free to use either codepoint to indicate ECT, on a packet-by-packet basis. The Not-ECT codepoint '00' indicates a packet that is not using ECN and the CE codepoint '11' is set by a router to indicate congestion to the end nodes.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Header Length				Reserved				C	E	U	A	P	R	S	F
								W	C	R	C	S	S	Y	I
								R	E	G	K	H	T	N	N

Figure 3.1 Addition of ECN to TCP Reserved Field.

Per specifications, two bits (bit 6 and 9) in the reserved field of the TCP header are used for ECN, as shown in Figure 3.1 [6].

Thus, ECN uses the ECT and CE flags in the IP header for signaling between routers and connection endpoints, and uses the Congestion Window Reduced (CWR) and ECN-Echo (ECE) flags in the TCP header for TCP-endpoint to TCP-endpoint signaling. For a TCP connection, a typical sequence of events in an ECN-based reaction to congestion is [6]:

- An ECT codepoint is set in packets transmitted by the sender to indicate that ECN is supported by the transport entities for these packets.
- An ECN-capable router detects impending congestion and detects that an ECT codepoint is set in the packet it is about to drop. Instead of dropping the packet, the router chooses to set the CE codepoint in the IP header and forwards the packet.
- The receiver receives the packet with the CE codepoint set, and sets the ECE flag in its next TCP ACK sent to the sender.
- The sender receives the TCP ACK with ECE set, and reacts to the congestion as if a packet had been dropped.
- The sender sets the CWR flag in the TCP header of the next packet sent to the receiver to acknowledge its receipt of and reaction to the ECE flag.

3.2 ECN Initialization

The negotiation for using ECN by the TCP transport entities and the use of the ECN-Echo and CWR flags is [6]:

- In the TCP connection setup phase, the source and destination TCPs exchange information about their willingness to use ECN.
- Subsequent to the completion of this negotiation, the TCP sender sets an ECT codepoint in the IP header of data packets to indicate to the network that the transport is capable and willing to participate in ECN for this packet.
- This indicates to the routers that they may mark this packet with the CE codepoint, if they would like to use that as a method of congestion notification.
- If the TCP connection does not wish to use ECN notification for a particular packet, the sending TCP sets the ECN codepoint to not-ECT, and the TCP receiver ignores the CE codepoint in the received packet.

Further details regarding the process of initialization as well as subsequent actions for ECN implementation on TCP/IP can be found in additional publications [6].

4. Implementation Of Simulation

4.1 Network Scenario for Simulation

Two network scenarios have been used for ECN performance analysis: LAN and WAN.

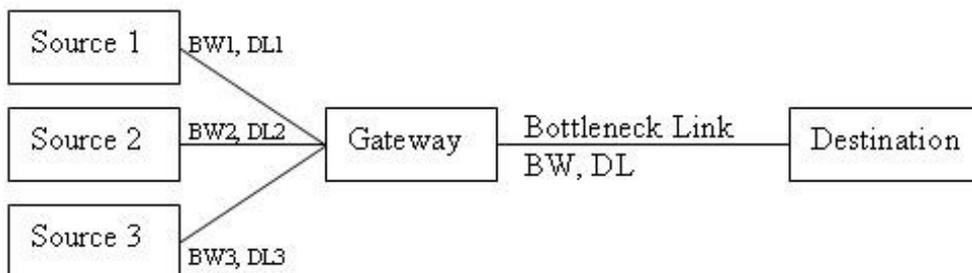


Figure 4.1 Simulation Network Scenario.

The block diagram in Figure 4.1 shows a typical network scenario, in which there are three sources, one bottleneck link common to all and one destination.

Parameter Type	Parameter Name (as per fig.4.1)	LAN	WAN
Bandwidth Parameter (Mbps)	BW1	100	100
	BW2	100	100
	BW3	100	100
	BW	10/100	0.5/1
Delay Parameter (ms)	DL1	1	1
	DL2	1	1
	DL3	1	1
	DL	1	10

Table 4.1: Typical Network Parameters for Simulation

As shown in Table 4.1, the LAN and WAN scenarios have different delay and bandwidth parameters chosen per characteristics of the two types of networks.

4.2 Performance Indicators

The simulation for different network scenarios and parameter values has been analyzed for the following performance indicators:

- Total number packet drops over the simulation period.
- Congestion window size variation in the sending nodes (sources)
- Overall data throughput
- Queue size variation in the bottleneck links
- Variation of these indicators at different packet size, queue delay, buffer size etc.

5. Simulation Results and Discussion

In all simulations, the buffer size at the source and destination nodes is constant at 50K Bytes.

5.1 Throughput

The data throughput has been calculated using the script '*throughput.pl*' [8]. A granularity of 1 second has been taken to calculate the throughput from the NS2 trace files generated by each instance of the simulation. An explanation of the trace format and its different fields can be found in other publications [9].

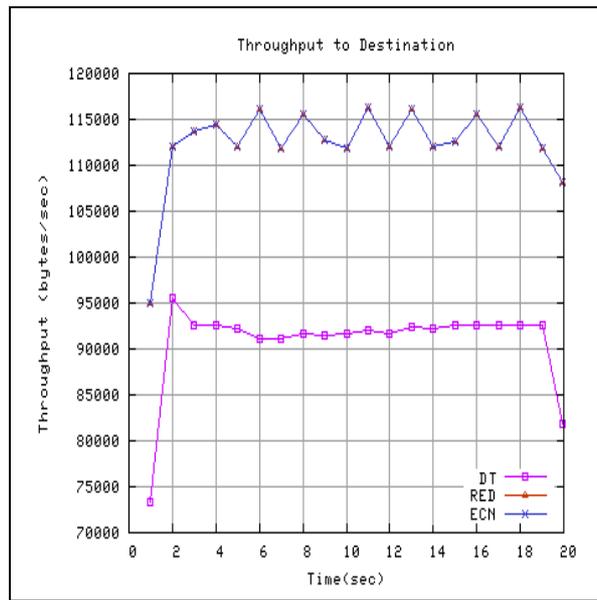


Fig. 5.1: Overall Throughput to Destination Node, WAN scenario, gateway buffer size 5K Bytes

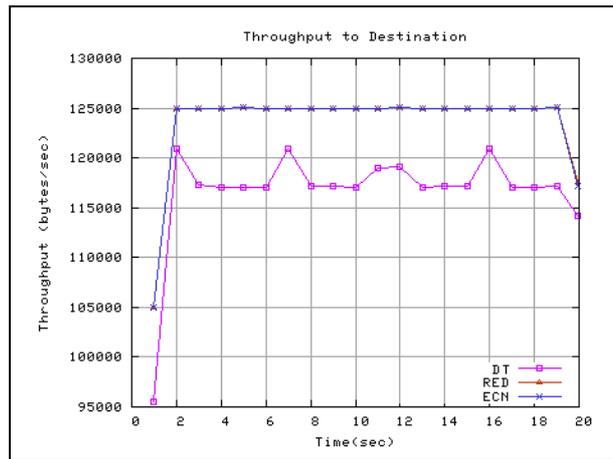


Fig. 5.2 Overall Throughput to Destination Node, WAN scenario, gateway buffer size 30K Bytes

The results in Fig. 5.1 and Fig. 5.2 indicate that the overall throughput for DropTail is much lower than both RED and ECN for lower values of gateway buffer size. Naturally, as we increase gateway buffer size, the performance of DropTail approaches RED and ECN as congestion and resultant packet drop decrease. RED and ECN have comparable throughput regardless of gateway buffer size.

5.2 Number of Packets Dropped

The number of packets dropped using DropTail, RED and ECN for WAN network scenario with different buffer sizes at the gateway node was determined. The total number of packets transferred over the simulation period of 20 seconds was 106,000.

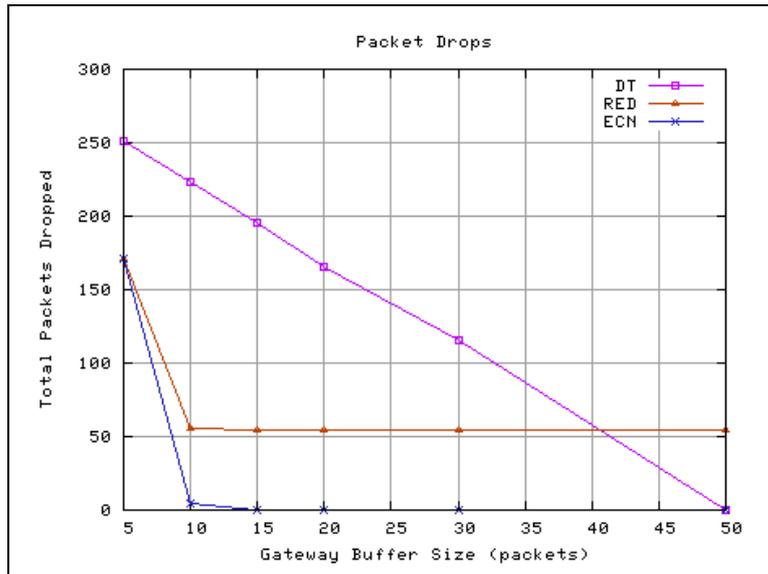


Fig. 5.3: Number of Packet Drops, WAN Scenario

The results in fig.5.3 indicate that the proportion of packet drops is significantly improved using ECN over RED, while the performance of DropTail is significantly worse than both other methods.

5.3 Gateway Bandwidth

This gateway bandwidth utilization is the overall packet forwarding rate at the gateway. This is calculated as the total kilobytes of data departing from the gateway queue per second.

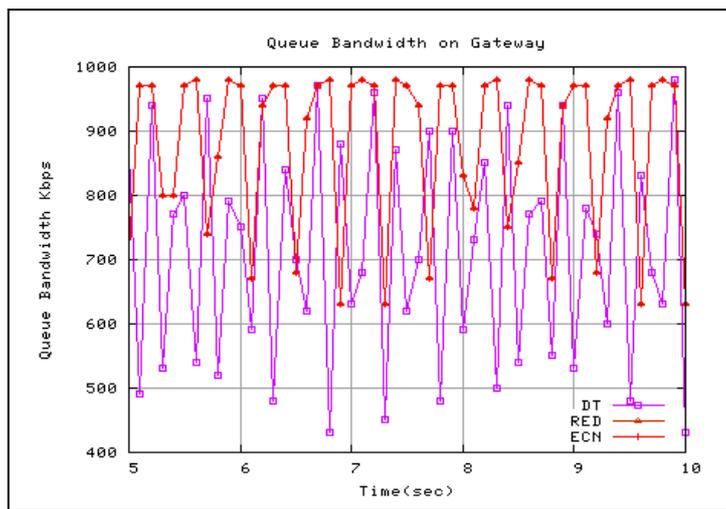


Fig 5.4: Gateway Bandwidth Utilization, WAN Scenario, Gateway Buffer Size 5K Bytes

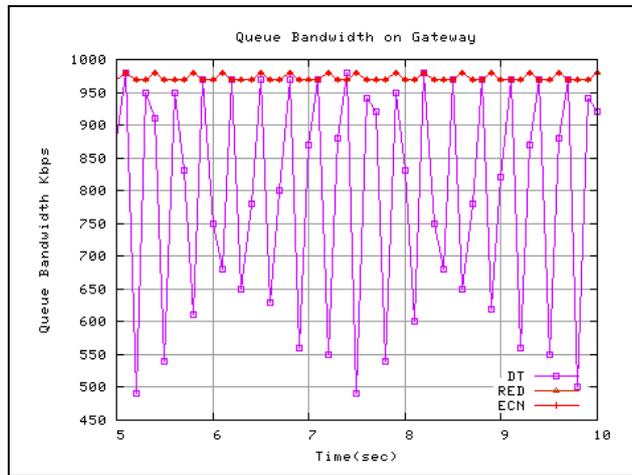


Fig 5.5: Gateway Bandwidth Utilization, WAN scenario, Gateway Buffer Size 10K Bytes

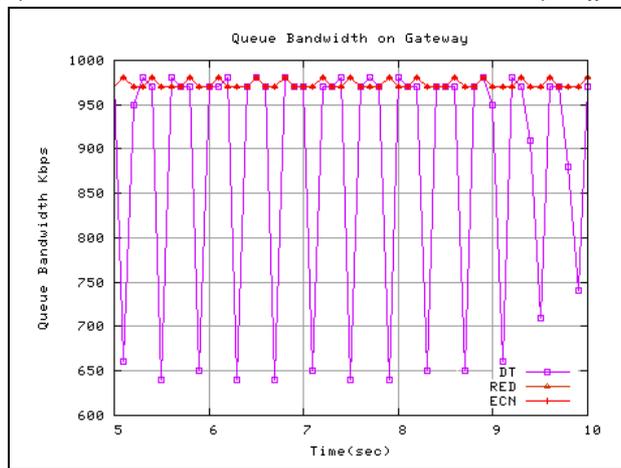


Fig 5.6: Gateway Bandwidth Utilization, WAN scenario, Gateway Buffer Size 20K Bytes

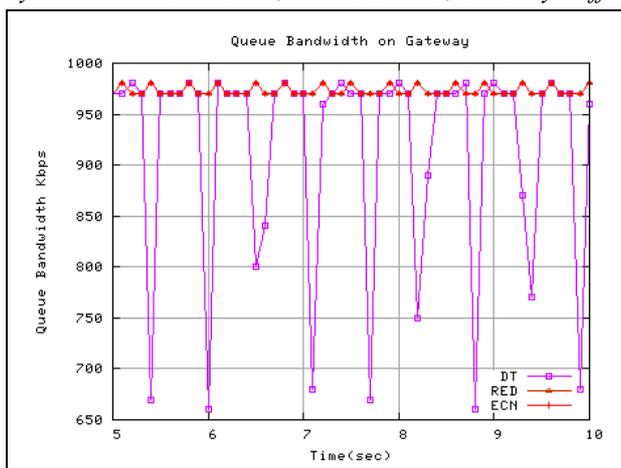


Fig 5.7: Gateway Bandwidth Utilization, WAN Scenario, Gateway Buffer Size 30K Bytes

The results in fig. 5.4 through 5.7 show that for WAN scenario, for low buffer size on the gateway (i.e. at higher congestion) the gateway bandwidth utilization of RED and ECN are comparable and both are much better and consistent than DropTail. The performance of DropTail naturally improves for higher gateway buffer size but it is still inconsistent as it involves rapidly fluctuating rates from sources (changing *congestion window sizes* due to drops).

5.4 Congestion Window Size Variation

The variation of the congestion window size (*cwnd*) on the sources of traffic for different scenarios and values of the gateway buffer size was determined.

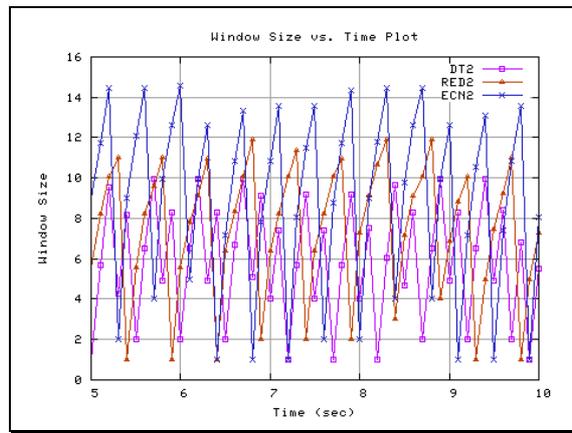


Fig 5.8: Congestion Window Size Variation, WAN Scenario, Gateway Buffer Size 5K Bytes

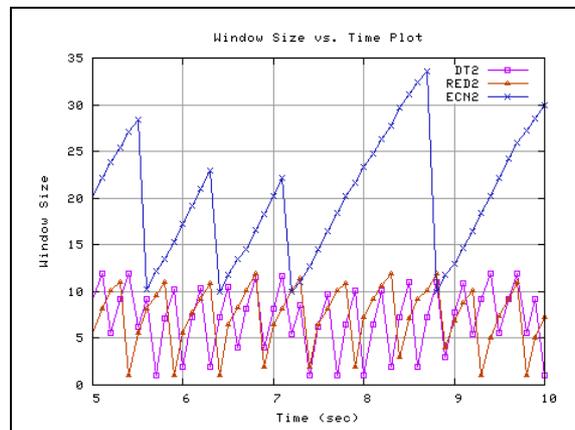


Fig 5.9: Congestion Window Size Variation, WAN Scenario, Gateway Buffer Size 10K Bytes

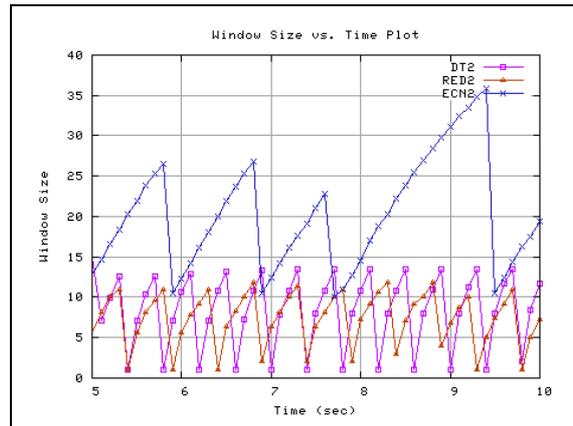


Fig 5.10: Congestion Window Size Variation, WAN Scenario, Gateway Buffer Size 20K Bytes

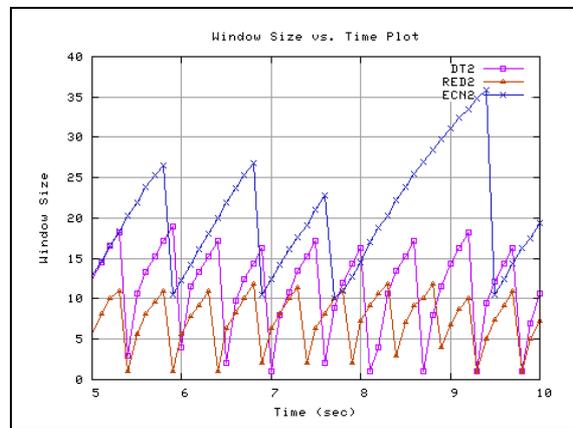


Fig 5.11: Congestion Window Size Variation, WAN Scenario, Gateway Buffer Size 30K Bytes

The results in Fig. 5.8 through Fig 5.11 show that the variation of $cwnd$ at the source behaves differently for different queue management techniques. It is more stable and optimized for ECN compared to both RED and DropTail. For ECN, since packet drops are significantly lowered, the $cwnd$ rarely goes to the slow-start phase whereas it frequently does so for RED and DropTail. Thus, as packet drops are minimized in ECN, direct positive consequences such as more optimized gateway bandwidth utilization and better consistency at the sources in terms of $cwnd$ variation are obtained. This increased consistency in $cwnd$ also means that there is less jitter in the network and the network resources are more optimally utilized.

5.5 Bottleneck Queue Size Variation

Queue size variation at the bottleneck node for different scenarios and values of the gateway buffer size was determined.. The results illustrate the accumulated queue size measured every 0.1 seconds at the gateway.

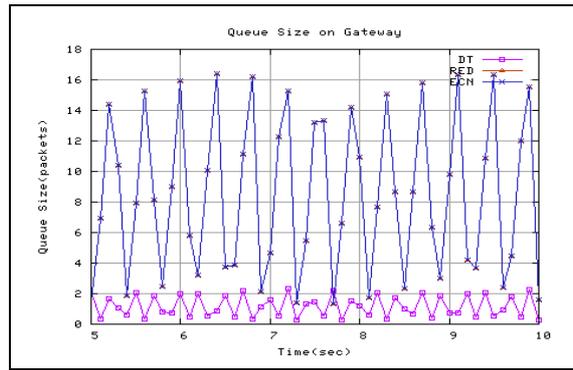


Fig. 5.12: Gateway Queue Size Variation, WAN scenario, Gateway Buffer Size 5K Bytes

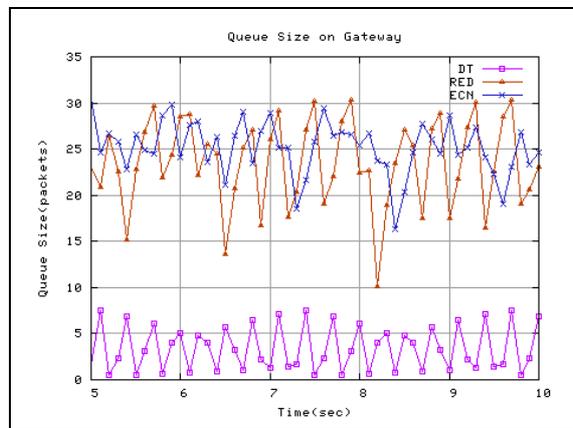


Fig. 5.13: Gateway Queue Size Variation, WAN scenario, Gateway Buffer Size 10K Bytes

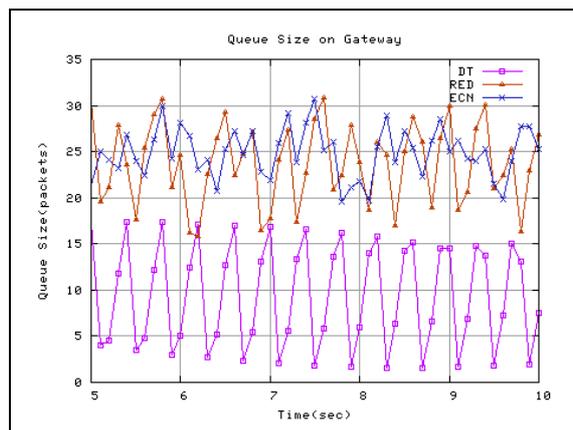


Fig. 5.14: Gateway Queue Size Variation, WAN scenario, Gateway Buffer Size 20K Bytes

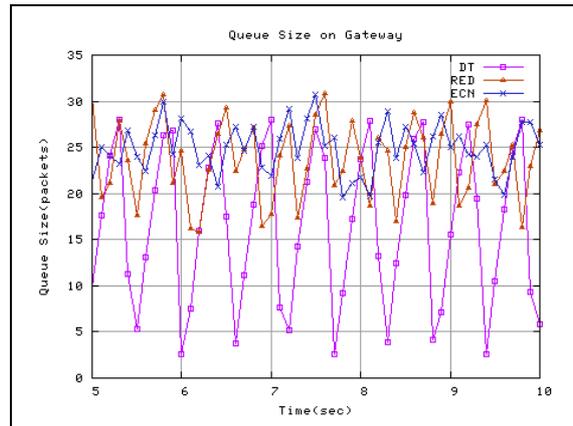


Fig. 5.15: Gateway Queue Size Variation, WAN scenario, Gateway Buffer Size 30K Bytes

The results in fig 5.12 through fig. 5.15 illustrate a pattern similar to the one for gateway bandwidth utilization. Both RED and ECN show similar variations with ECN having a better consistency and DropTail has much lower queue utilization at the gateway. This is because, as the number of packet drops is high, the source nodes continuously throttle their rates and the queue at gateway becomes underutilized.

6. Conclusion

From the results of the simulations it has been clearly observed that when all network prerequisites are met ECN reduces packet drops and thereby optimizes network resource utilization and data throughput.

From even the very basic network scenario used as the network topology for this analysis, many important characteristics of DropTail, RED and ECN have been clearly demonstrated. Obviously, DropTail is the least efficient of the three and it is such a straightforward queue management algorithm it does not even qualify as congestion management. RED is a more intelligent queue management system and can be termed as a congestion management system. ECN goes beyond RED by using packet marking, rather than dropping, to notify network nodes of impending congestion. It is an intelligent feedback system that attempts to optimize the network throughput and resource utilization while minimizing packet drops. Still, it is understandable that even ECN can not totally eliminate packet drops.

REFERENCES

- [1] Altman and Jimenez, NS for Beginners, <http://www-sop.inria.fr/maestro/personnel/Eitan.Altman/COURS-NS/n3.pdf>
- [2] Sally Floyd: A Proposal to add Explicit Congestion Notification (ECN) to IP, RFC2481 (<ftp://ftp.isi.edu/in-notes/rfc2481.txt>)
- [3] Sally Floyd: TCP and Explicit Congestion Notification, Lawrence Berkley Laboratory, ACM Computer Communication Review, V. 24 N. 5, October 1994, p. 10-23, (http://www.icir.org/floyd/papers/tcp_ecn.4.pdf)

- [4] Sally Floyd: The Addition of Explicit Congestion Notification (ECN) to IP, RFC 3168, (<ftp://ftp.isi.edu/in-notes/rfc3168.txt>)
- [5] Sally Floyd's ECN page: (<http://www.icir.org/floyd/ecn.html>), contains many other related links not mentioned here
- [6] Steven H. Low, Fernando Paganini and John C. Doyle: Internet Congestion Control, IEEE Control System Magazine, February 2002
- [7] The NS2 Network Simulator program: <http://www.isi.edu/nsnam/ns>
- [8] VINT Project, NS2 Manual, <http://www.isi.edu/nsnam/ns/ns-documentation.html>
- [9] W. Stevens: TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms, RFC 2001 (<http://www.faqs.org/ftp/rfc/pdf/rfc2001.txt>)