Traffic Management using Multilevel Explicit Congestion Notification[★]

Arjan Durresi, Mukundan Sridharan, Chunlei Liu, Mukul Goyal
Department of Computer and Information Science
The Ohio State University
Columbus, OH 43210-1277, USA
durresi@cis.ohio-state.edu
and
Raj Jain

Raj Jain is now at
Washington University in Saint Louis
Jain@cse.wustl.edu
http://www.cse.wustl.edu/~jain/

ABSTRACT

Congestion remains the main obstacle to the Quality of Service on the Internet. We believe that a good solution to Internet congestion should optimally combine congestion signaling from network and source reaction, having as main goals: minimum losses and delays, maximum network utilization, fairness among flows and last but not least scalability of the solution.

In this paper we present a new traffic management scheme based on an enhanced Explicit Congestion Notification (ECN) mechanism. In particular we used multilevel ECN, which conveys more accurate feedback information about the network congestion status than the current two levels ECN. We have designed a TCP source reaction that takes advantage of the extra information about congestion and tunes better its response to the congestion than the current schemes. Our analysis and simulations results shows that our scheme performed better than the current ones, having less losses, better network utilization, better fairness, and the solution is scalable.

1. INTRODUCTION

There is an immense demand for quality of service (QoS) in the Internet. One key element of quality of service is traffic management. Since the network traffic is bursty, it is difficult to make any QoS guarantees without proper control of traffic. Currently, Internet Protocol (IP) has only minimal traffic management capabilities. The packets are dropped when the queue exceeds the buffer capacity. The transmission control protocol (TCP) uses the packet drop as a signal of congestion and reduces its load [18]. While in the past, this strategy has worked satisfactorily, now we need better strategies for two reasons [12, 15, 16]. First, the bandwidth of the networks as well as the distances are increasing. For very high distance-bandwidth product networks, packet drop is not the optimal congestion indication. Several megabytes of data may be lost in the time required to detect and respond to packet losses. Therefore, a better strategy for traffic management in IP networks is required. Second, a large part of the traffic, particularly, voice and video traffic does not use TCP. Continuous media traffic uses User Data Protocol (UDP). The proportion of UDP traffic is increasing at a faster pace than TCP traffic. The UDP traffic is congestion insensitive in the sense that UDP sources do not reduce their load in response to congestion [5].

Despite the fact that a number of schemes have been proposed for congestion control, the search for new schemes continues [1, 3, 6, 7, 8, 9 - 17]. The research in this area has been going on for at least two decades. There are two reasons for this. First, there are requirements for congestion control schemes that make it difficult to get a satisfactory solution. Second, there are several network policies that affect the design of a congestion scheme. Thus, a scheme developed for one network, traffic pattern, or service requirements may not work on another network, traffic pattern, or service requirements. For example, many of the schemes developed in the past for best-effort data networks will not work satisfactorily for multi-class IP networks.

Recognizing the need for a more direct feedback of congestion information, the Internet Engineering Task Force (IETF) has come up with Explicit Congestion Notification (ECN) method for IP routers [1, 2]. A bit in the IP header is set when the routers are congested. ECN is much more powerful than the simple packet drop indication used by existing routers and is more suitable for high distance-bandwidth networks. In this paper we present some enhancement to ECN based on multilevel ECN. Our results show that Multilevel ECN (MECN)

[★] This work was supported in part by grants from TRW, Honeywell, OAI, Cleveland, Ohio and NSF CISE grant #9980637

Proceedings of the 5th World Multiconference on Systemics, Cybernetics and Informatics SCI'2001, ABR over the Internet, Won best paper award, Orlando, FL, July 22-25, 2001, pp. 12-17

Raj Jain is now at Washington University in Saint Louis, jain@wustl.edu

improves considerably the congestion control. The remaining of the paper is organized as follows: In Section 2 we present the MECN including the router marking and dropping policy, receiver feedback and the TCP source response. In Section 3 we present the results of our simulations with MECN and compare them with ECN results. The conclusions of the study are in Section 4.

2. MULTILEVEL ECN (MECN)

Marking the bits at the router

The current proposal for ECN uses two bits in the IP header (bits 6 and 7 in the TOS octet in Ipv4, or the Traffic class octet in Ipv6) to indicate congestion. The first bit is called ECT (ECN-Capable Transport) bit. This bit is set to 1 in the packet by the traffic source if the source and receiver are ECN capable. The second bit is called the CE (congestion Experienced) bit. If the ECT bit is set in a packet, the router can set the CE bit in order to indicate congestion.

The two bits specified for the purpose of ECN can be used more efficiently to indicate congestion, since using two bits we can indicate 4 different levels. If non ECN-capable packets are identified by the bit combination of ‘00’, we have three other combinations to indicate three levels of congestion. In our scheme the bit combination ‘01’- indicates no congestion,’10’-indicates incipient congestion and ‘11’- indicates moderate congestion. Packet drop occurs only if there is severe congestion in the router and when the buffer over flows. So with packet-drop we have four different levels of congestion indication and appropriate action could be taken by the source TCP depending on the level of congestion. The four levels of congestion are summarized in Table 1.

The marking of CE, ECT bits is done using a multilevel RED scheme. The RED scheme has been modified to include another threshold called the mid_thresh, in addition to the min_threshold and max_threshold. If the size of the average queue is in between min_th and mid_th, there is incipient congestion and the CE, ECT bits are marked as ‘10’ with a maximum probability of P1max. If the average queue is in between mid_th and max_thresh, there is moderate congestion and the CE, ECT bits are marked as ‘11’ with a maximum probability P2max. If the average queue is above the max_thresh all packets are dropped. The marking policy is shown in Fig. 1.

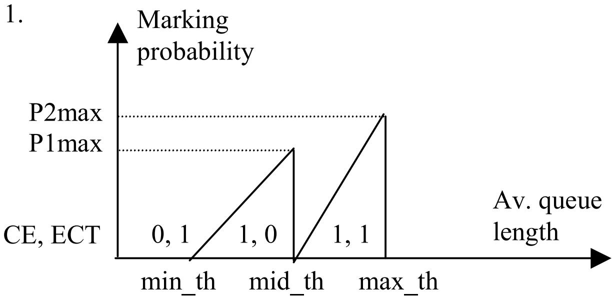

Figure 1. Marking at routers

Table 1. Router response to congestion: marking of CE and ECT bits and packet dropping

CE bit	ECT bit	Congestion State
0	1	No Congestion
1	0	Incipient congestion
1	1	Moderate congestion
Packet drop		Severe congestion

Feed back from Receiver to Sender

The receiver reflects the bit marking in the IP header, to the TCP ACK. Since we have three levels of marking instead of 2-level marking in the traditional ECN, we make use of 3 combination of the 2 bits 8, 9 (CWR, ECE) in the reserved field of the TCP header, which are specified for ECN. Right now the bit combination ’00’ indicates no congestion and ‘01’ indicates congestion. Again, these 2 bits are just going to reflect the 2 bits in the IP header. The packet drop is recognized using traditional ways, by timeouts or duplicate ACKs.

The receiver marks the CWR, ECE bits in the ACKs as "01' if the received packet has a CE, ECT bits marked by the router as '10'. When a packet with CE, ECT bits marked as '11' is received, the receiver marks CWR, ECE bits in ACKs as '11'. If the received packet has a CE. ECT bits marked as '00' or '01', the receiver marks CWR, ECE bits of the ACKs as ‘00’. The marking in the ACKs CWR, ECE bits is shown in Table 2.

In the current ECN standard the CWR bit has the possibility of being set only in packets from source to the receiver and the receiver stops reflecting the ECN bits if it receives a packet with CWR set. But in our scheme the CWR is used in both directions. So we made the following changes to the TCP. If an end-user receives a ACK packet with bit 8 in the reserved field set (CWR), it reduces its ‘cwnd’ proportionately, if the packet is not a ACK packet then it stops reflecting the congestion levels in the ACK packets.

Response of TCP source

We believe that the marking of ECN should not be treated as the same way as a packet drop, since ECN indicates just the starting of congestion and not actual congestion and the buffers still have space. And now with multiple levels of congestion feedback, the TCP’s response needs to be refined.

We have implemented the following scheme: When there is a packet-drop the ‘cwnd’ is reduced by α_3 = 50%. This done for two reasons: First, a packet-drop means severe congestion and buffer overflow and some severe actions need to be taken. Second, to maintain backward compatibility with routers which don’t implement ECN.

For other levels of congestion, such a drastic step as reducing the ‘cwnd’ as half is not necessary and might make the flow less vigorous. When there is no congestion,

Table 2. Receiver marking of CWR and ECE bits

CWR bit	ECE bit	Congestion
0	0	No Congestion or non-ECN capable
0	1	Incipient congestion
1	1	Moderate congestion

the 'cwnd' is allowed to grow additively as usual. When the marking is '10'(incipient congestion), 'cwnd' is decreased by α_1 %. When the marking is '11'(moderate congestion) the 'cwnd' is decreased multiplicatively not by a factor of 50% (as for a packet –drop), but by a factor α_2 % less than 50% but more than α_1. In Table 3 are shown the TCP source responses and the value of parameters α_x we have implemented. In future work we will study the values of parameters α_x. Another method could be to decrease additively the window, when the marking is '10'(incipient congestion), instead of maintaining the window. This again will be analyzed in future study.

3. SIMULATION

In order to compare the present ECN with our multi-level ECN scheme, we carried out a set of simulations using the NS simulator [ ]. The RED queue in the NS has been modified to include the mid_thresh, in addition to the min_threshold and max_threshold. The marking policy is shown in Fig. 1 and is explained in Section 2.

The TCP in the ns simulator is also modified according to our algorithm. The receiver, just reflects the markings in the IP header, in the experimental field of the TCP header. The sender reduces its congestion window by 20% if it gets a mild congestion marking and reduces the window by 40%, if it gets a heavy congestion marking. If there is any timeout or duplicate acks (packet loss) the TCP reduces the window by 50%. When the TCP sender sends the congestion window reduced (CWR) signal, the receiver stops echoing the level of ECN, which it marked first. For example, suppose if there is congestion in the router and it starts marking packets in the next level. The receiver gets packets and starts echoing that particular level of ECN in all acks. Suppose if the congestion makes into next level, before the receiver gets a congestion window reduced (CWR) signal, the receiver remember, which level was marked first and stops echoing that level and starts echoing the next level of ECN. The connection establishment phase and the ECN negotiation are not modified.

For simplicity, the max Probability of dropping , for both levels of ECN are kept the same, P1max = P2max. Also for the same reason we have applied for MECN max_th = 2 mid_th, and mid_th = 2 min_th and for simple ECN max_th = 2 min_th. The aim of the simulation is not to fix the best parameter of the RED queue, but to illustrate the advantage of multi-level ECN. Further study is needed to optimize these parameters.

Table 3. TCP source response

Congestion State	cwnd change
No congestion	Increase 'cwnd' additively
Incipient congestion	Decrease multiplicatively by $\alpha_1 = 20\%$
Moderate congestion	Decrease multiplicatively by $\alpha_2 = 40\%$
Severe congestion	Decrease multiplicatively by $\alpha_3 = 50\%$

Simulation Configuration

For all our simulations we used the following configuration. A Number of sources S1, S2, S3.., Sn are connected to a router R1 through 10Mbps, 2ms delay links. Router R1 is connected to R2 through a 1.5Mbps, 20ms delay link and a number of destinations D1, D2, D3.., Dn are connected to the router R2 via 10Mbps 4ms delay links. The link speeds are chosen so that the congestion will happen only between routers R1 and R2 where our scheme is tested. In Fig. 2 is shown the simulation configuration.

With this configuration the fixed round trip time, including the propagation time and the transmission time at the routers is 59 ms. Changing the propagation delay between the source and router R1 gives us configurations of different RTT. An FTP application runs on each source. Reno TCP is used as the transport agent. (The modifications were made to the Reno TCP). The packet size is 1000 bytes and the acknowledgement size is 40 bytes.

Simulation Scenarios

With the basic configuration described above the following simulation scenarios were used to test our scheme.

1. Two connection with same RTT
2. Two overlapping connections with same rtt, the first source starts at 0 second and stops at 9.5 seconds and the second starting at 0.5 second and stopping at 10 seconds.
3. Ten connections with same RTT.
4. Ten overlapping connections with same RTT, each connection starting 0.3 seconds after the previous one.
5. Two connections with different RTT
6. Five connections with different RTT.

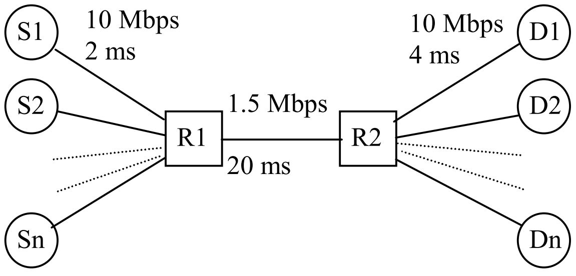

Figure 2. Simulation configuration

In Fig. 3a. are shown the instantaneous and average queue lengths in number of packets for the case of two sources with the same RTT applying simple ECN. In Fig. 3b are the results for the same configuration applying our scheme for MECN. It is clear that in case of MECN the queue length converge faster and with less oscillations compared to ECN case.

In Fig. 4 there are compared the link efficiencies obtained with simple ECN and MECN in the configuration with two sources with the same RTT for different min_threshold in number of packets. As shown MECN not only has improved the link efficiency for all values of min_threshold, but the best link efficiency in case of MECN is reached for a lower value of min_threshold, that means with less delays introduced to the packets.

In Fig. 5 we compare the throughput of one individual source applying ECN and MECN in the case of ten sources with overlapping connections and the same RTT configuration. Again MECN provides better throughput than simple ECN.

In Fig. 6 we compare the losses in number of packets with simple ECN and MECN for the configuration with two sources. As shown in case of MECN there are less losses than with simple ECN and MECN reaches the point of zero loss with less min_threshold than ECN.

In Fig. 7 again there are compared the losses between ECN and MECN for the configuration with five sources with different RTT. In this case the improvements of MECN compared to ECN is considerable in having less losses and reaching the point of zero losses with much less min_threshold, that means with mach less delay introduced to packets.

All our simulation experiments show that Multi-level ECN works better than simple ECN as a congestion control scheme for TCP. We plan to study in the future the influence of different parameters involved for example min, mid, max thresholds, P1max, P2max, Ax and other open issues. Also we plan to simulate more complex configurations to study better the advantages of MECN in more realistic scenarios.

4. CONCLUSIONS

In this paper we have introduced Multi-level ECN (MECN), a new congestion control scheme for TCP based on the ECN proposal. With MECN the routes can send more detailed information about the congestion to the TCP source and destination. Having more detailed information about the congestion make possible for the TCP sources to have a better tuned response to the congestion. As result the MECN congestion control scheme converge faster and with less losses than simple ECN.

In our proposal for MECN we make use of the same bits in IP and TCP header already used by ECN, so our proposal is compatible with the accepted standards.
All our simulation results show that MECN improves the QoS parameters such as throughput, link utilization, delay, losses, and queue oscillation compared to ECN scheme.

Even though further study is needed, we believe that MECN is a step forward in the right direction to deal with Internet congestion.

REFERENCES

[1] K. Ramakrishnan and S. Floyd, ``A Proposal to add Explicit Congestion Notification (ECN) to IP," *RFC 2481*, January 1999

[2] Sally Floyd, ``TCP and Explicit Congestion Notification," *Computer Communications Review*, Vol. 24, No. 5, October 1994, pp. 10-23.

[3] Sally Floyd and Van Jacobson, ``Random Early Detection gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, Vol. 1, No. 4, August 1993, pp. 397-413.

[4] Sally Floyd and Van Jacobson, ``On Traffic Phase Effects in Packet-Switched Gateways," *Internetworking: Research and Experience*, Vol. 3, No. 3, September 1992, pp. 115-156.

[5] Sally Floyd, ``Connections with Multiple Congested Gateways in Packet-Switched Networks Part 1: One-way Traffic," *Computer Communications Review*, Vol. 21, No. 5, October 1991, pp. 30-47.

[6] David D. Clark and Wenjia Fang, ``Explicit allocation of best-effort packet delivery service," *IEEE/ACM Trans. on Networking*, Vol. 6, No. 4, August 1998, pp. 362-373.

[7] W. Feng, D. Kandlur, D. Saha and K. Shin, ``Blue: A New Class of Active Queue Management Algorithms," University of Michigan, *Technical Report UM-CSE-TR-387-99*, 1999.

[8] S. Kalyanaraman, R. Jain, S. Fahmy R. Goyal and B. Vandalore, ``The ERICA Switch Algorithm for ABR Traffic Management in ATM Networks," Submitted to *IEEE/ACM Trans. on Networking*, November 1997, available at http://www.cis.ohio-state.edu/~jain/papers/erica.htm

[9] Sally Floyd and Kevin Fall, ``Promoting the Use of End-to-End Congestion Control in the Internet," *IEEE/ACM Trans. on Networking*, August 1999.

[10] Sally Floyd and Kevin Fall, ``Router Mechanisms to Support End-to-End Congestion Control," *Unpublished Manuscript*, http://www-nrg.ee.lbl.gov/floyd/papers.html

[11] M. Mathis, J. Semke, J. Mahdavi and T. Ott, ``The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm," *Computer Communications Review*, Vol. 27, No. 3, July 1997, pp. 67-82.

[12] D. Chiu and R. Jain, ``Analysis of the Increase/Decrease Algorithms for Congestion

Avoidance in Computer Networks," *Journal of Computer Networks and ISDN Systems*, Vol. 17, No. 1, June 1989, pp. 1-14.

[13] Sally Floyd and Tom Henderson, ``The NewReno Modification to TCP's Fast Recovery Algorithm,", *Internet Draft - Work in Progress*, Feburary 1999.

[14] M. Mathis, J. Mahdavi, S. Floyd and A. Romanow, ``TCP Selective Acknowledgment Options, " *RFC 2018*, October 1996.

[15] K. K. Ramakrishnan and R. Jain, ``A binary feedback scheme for congestion avoidance in computer networks," *ACM Transactions on Computer Systems*, Vol. 8, No. 2, May 1990, pp. 158-181.

[16] R. Jain, S. Kalyanaraman, R. Viswanathan, ``Rate Based Schemes: Mistakes to Avoid." *AF-TM 94-0882*, September 1994.

[17] R. Jain, S. Kalyanaraman, and R. Viswanathan, ``Ordered BECN: Why we need a timestamp or sequence number in the RM Cell," *ATM Forum/94-0987*, October 1994.

[18] W. Stevens, ``TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," *RFC 2001*, January 1997.

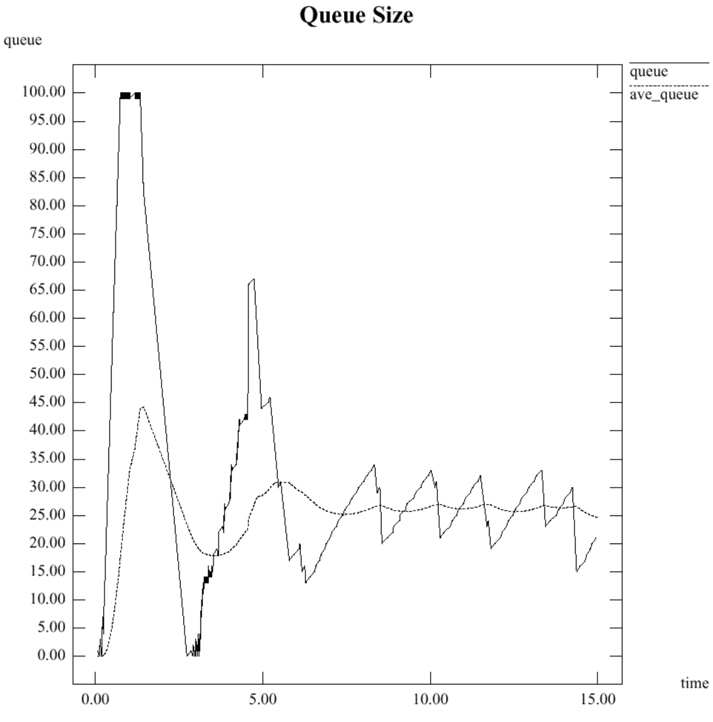

Figure 3a. Queue size in two sources simulations using ECN

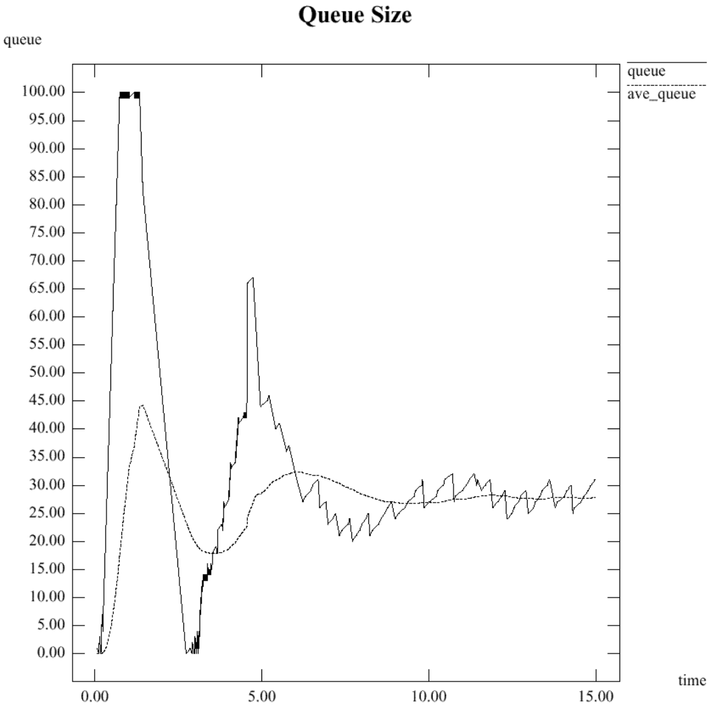

Figure 3b. Queue size in two sources simulations using MECN

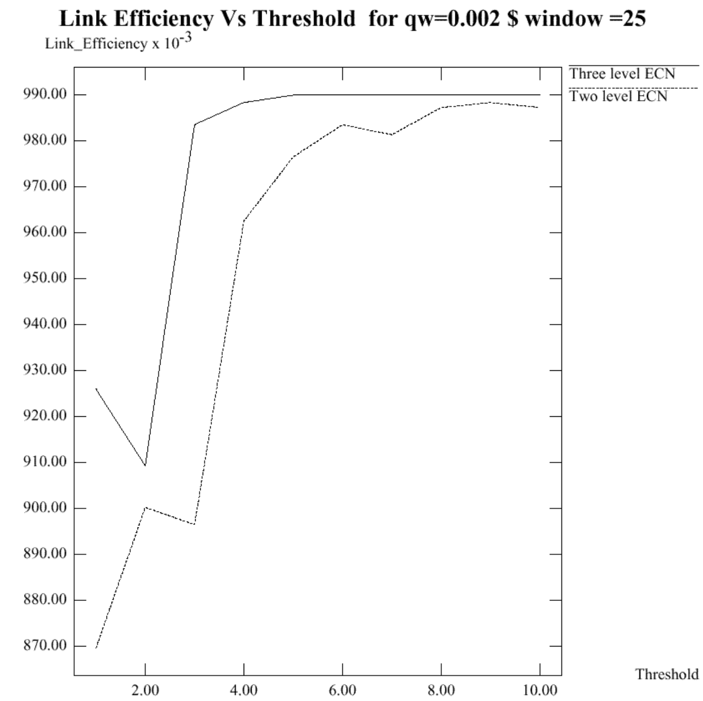

Figure 4. Link efficiency in two sources simulation

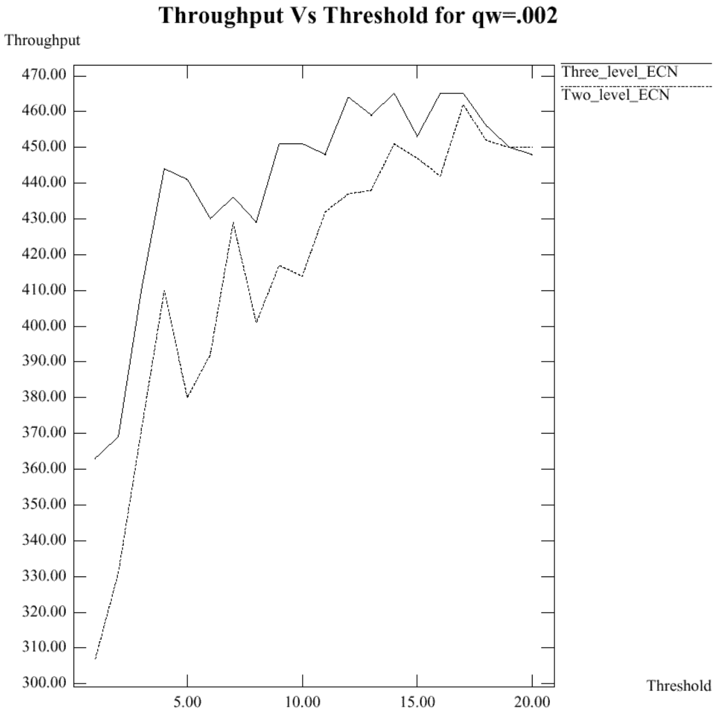

Figure 5. Throughput in ten sources simulation

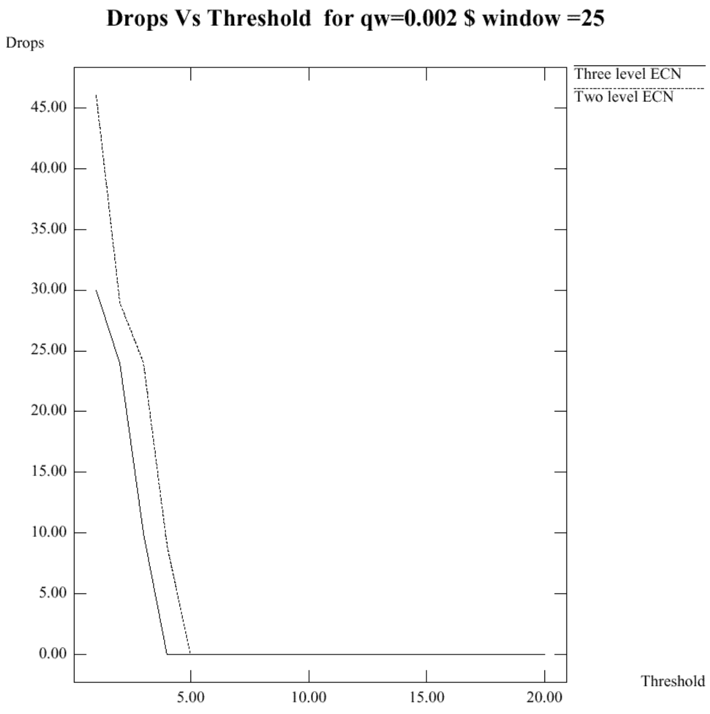

Figure 6. Packet drops in two sources simulation

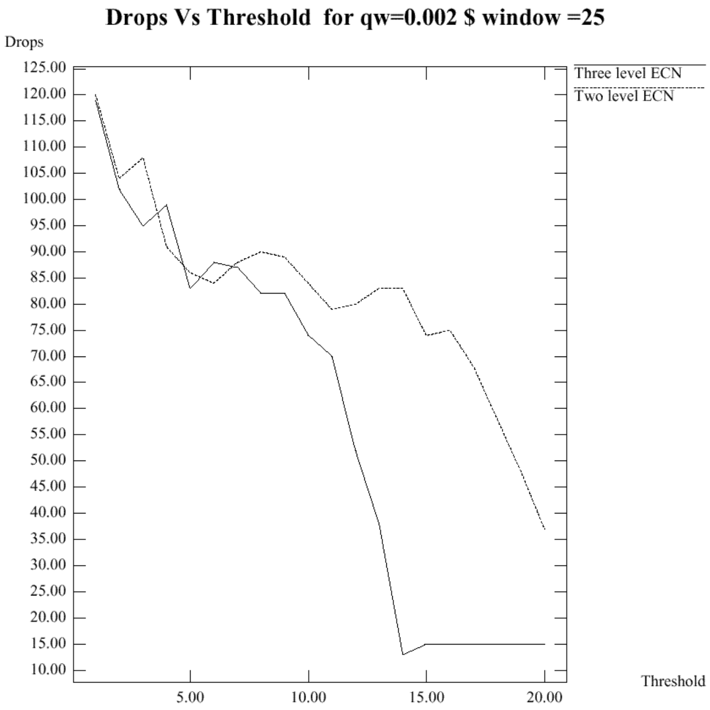

Figure 7. Throughput in ten sources simulation