Guidelines for optimizing Multi-Level ECN, using fluid flow based TCP model

Pierre-François Quet[a] Sriram Chellappan[a] Arjan Durresi[b]
Mukundan Sridharan[b] Hitay Özbay[a] Raj Jain[c]

[a]Dept. of Electrical Engineering
[b]Dept. of Computer and Information Science
The Ohio State University
2015 Neil Avenue
Columbus, OH 43210, USA

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

ABSTRACT

Congestion avoidance on today's Internet is mainly provided by the combination of the TCP protocol and Active Queue Management (AQM) schemes such as the *de facto* standard RED (Random Early Detection). When used with ECN (Explicit Congestion Notification), these algorithms can be modeled as a feedback control system in which the feedback information is carried on a single bit. A modification of this scheme called MECN[1] was proposed, where the marking information is carried using 2 bits. MECN conveys more accurate feedback about the network congestion to the source than the current 1-bit ECN. The TCP source reaction was modified so that it takes advantage of the extra information about congestion and adapts faster to the changing congestion scenario leading to a smoother decrease in the sending rates of the sources upon congestion detection and consequently resulting in an increase in the router's throughput. A linearized fluid flow model already developed for ECN[2] is extended to our case. Using control theoretic tools we justify the performance obtained in using the MECN scheme and give guidelines for optimizing its parameters. We use `ns` simulations to illustrate the performance improvement from the point of better throughput and low level of oscillations in the queue.

Keywords: MECN, ECN, Fluid flow model, Congestion control, RED, AQM, QoS

1. INTRODUCTION

The bandwidth demand of the Internet is increasing at a rapid rate. Also, the bandwidth delay product of today's networks is very high, due to which several MegaBytes of data might be lost quickly if congestion occurs. Also, as the volume of Audio/Video traffic increases in the network, the requirement for Quality of Service (QoS) increases. It is highly difficult to guarantee QoS services to the end host if the network has congestion, as the delay in the networks fluctuates rapidly and a key requirement for QoS is less variation in delay. As a result, there is a pressing demand for efficient congestion control schemes in todays Internet as can be found in[3–6] , thus prompting refinement of existing congestion control schemes.

End-to-end congestion control in today's Internet is mainly due to the interaction between end users implementing the TCP protocol[7] and Active Queue Management schemes such as RED,[8] ECN.[9] ECN just marks one bit of information in each packet if marked. A proposed modification of ECN, namely Multi-Level Explicit Congestion Notification MECN[1] , is one where two bits per packet are marked, thus allowing us to indicate four different types of marking.

Emails: {quet.1, chellappan.1}@osu.edu, durresi@cis.ohio-state.edu, {sridharan.9, ozbay.1}@osu.edu, raj@nayna.com

In congestion-avoidance mode traditional TCP reduces its window size by 2 upon the detection of a marked packet (see[7]), which corresponds to an exponential decrease of the sending rates when congestion is detected. Such a system is thought to be harsh during stages of incipient congestion. The proposed modification of TCP is one where its congestion window is reduced differently depending on the type of mark an ACK packet carries: during severe congestion the source divides its congestion window by 2,which corresponds to halving the window size, but the main idea is to design a scheme in which for incipient or moderate congestion the congestion window is not halved and the reduction in window size should be milder, thus leading to a smoother decrease of the sources sending rates upon detection of a packet mark than the standard TCP, and consequently resulting in an increase in the system's throughput. In[1] , it was shown that MECN does improve the throughput at the router. However guidelines for proper tuning of the many parameters in it was still an issue. The remaining part of the paper is organized as follows: in Section 2 we present the MECN scheme including the router marking and dropping policy, receiver feedback and the TCP source response. In Section 3, we use a fluid-flow model similar to the one developed for RED-ECN in[2,10] which allows us to use control theoretic tools in order to analyze and validate the performance improvement and study stability margins of the proposed scheme compared to standard ECN. In Section 4, we use `ns` simulations to illustrate the performance improvement brought about by our new scheme and selective guidelines for parameter tuning is also given to optimize the system's performance. In Section 5, we present conclusions of the study and scope for future work.

2. MULTILEVEL EXPLICIT CONGESTION NOTIFICATION (MECN)

2.1. Marking the bits at the router

The current standard 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)[9,11] 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 we can indicate 4 different levels using two bits. If non ECN-capable packets are identified by the bit combination of '00', we have three other combinations to indicate three levels of congestion.

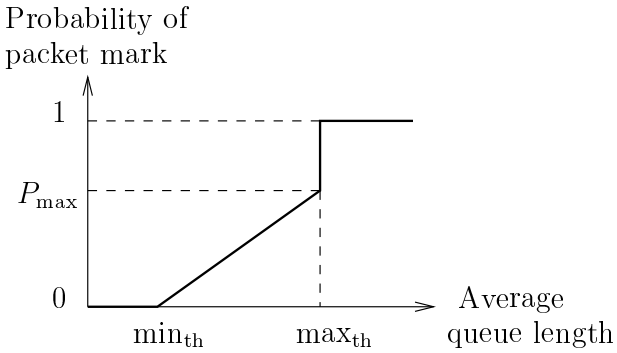
The marking of CE, ECT bits is done using a multilevel RED scheme. The RED scheme shown for comparison in Figures 1 and 2 has been modified to include another threshold called the mid_{th}, in addition to the max_{th} and min_{th}. If the size of the average queue is between min_{th} and mid_{th}, there is incipient congestion and the CE, ECT bits are marked as '10' with a probability p_1. If the average queue is between mid_{th} and max_{th}, there is moderate congestion and the CE, ECT bits are marked as '11' with a probability p_2 and the packets which did not get marked with '11' get marked with '10' with a probability p_1. If the average queue is above max_{th} all packets are marked with '11', and the packets are dropped if the buffer is full. The marking policy is shown in Figures 3 and 2 where $K = -C\log(1-\alpha)$ with α being the queue averaging weight and C being the outgoing link capacity*.

2.2. Feedback 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[9,11] , we make use of 3 combinations of the 2 bits 8, 9 (CWR, ECE) in the reserved field of the TCP header, which are specified for ECN. In traditional TCP the bit combination '00' indicates no congestion and '01' indicates congestion. Now 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 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 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 1.

*The low pass filter formulation is due to[10]



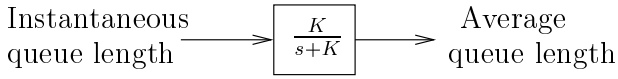


Figure 2. RED's queue averaging

Figure 1. Probability of marking a packet for ECN-RED

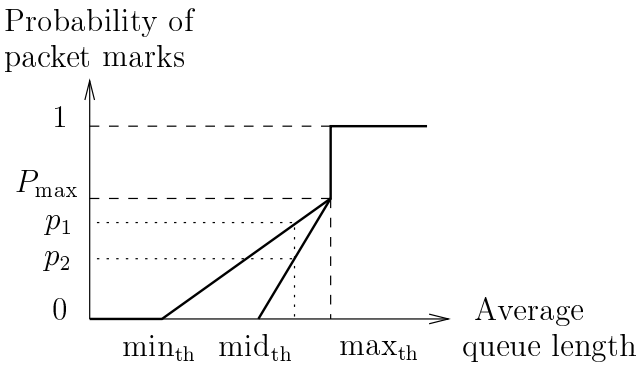


Figure 3. Probabilities of marking a packet for MECN

Table 1. Marking in the TCP ACKs CWR, ECE bits

CE bit	ECT bit	CWR bit	ECE bit
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

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.

2.3. Response of TCP source

In congestion-avoidance mode traditional TCP divides its congestion window size by 2 upon the detection of a marked packet or if a packet is dropped (see[7]), which corresponds to an exponential decrease in the sending rate. That is, a marking is treated the same as a dropping. We believe that the marking of ECN should not be treated the same way as a packet drop, since ECN indicates just the start of congestion, not actual congestion, and the buffers still have space. 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 divided by $\beta_3 = 2$. This done for two reasons: first, a packet-drop means severe congestion, buffer overflow, and some severe actions need to be taken; second, to maintain backward compatibility with routers which do not implement ECN.
- for other levels of congestion, such a drastic step as reducing the 'cwnd' by half is not necessary and might make the flow less vigorous. When there is no congestion, the congestion window is allowed to grow linearly as usual. When the marking is '01' the 'cwnd' is divided by β_1. When the marking is '11' the 'cwnd' is divided by a factor β_2. In Table 2 we show the TCP source response and the values of parameters we have implemented.

Table 2. Response to congestion of the modified TCP

CWR bit	**ECE bit**	**'cwnd' change**
0	0	'cwnd' increases linearly
0	1	'cwnd' divided by $\beta_1 = 5$
1	1	'cwnd' divided by $\beta_2 = 2.5$
Packet drop		'cwnd' divided by $\beta_3 = 2$

3. STABILITY AND PERFORMANCE ANALYSIS

In the following we study the stability and performance of the new system when the average queue length is between mid_{th} and max_{th}. The results obtained by our analysis are still valid even if the queue settles below mid_{th} as we will explain at the end of this section. In the following we ignore the TCP slow start and time out mechanisms, thus providing a model and analysis during the congestion avoidance mode only.

In TCP, the congestion window size ($W(t)$) is increased by one every round trip time if no congestion is detected, and is halved upon a congestion detection. This additive-increase multiplicative-decrease behavior of TCP has been modeled in[10] by the following equation (case of one TCP flow interacting with a single router)

$$dW(t) = \frac{dt}{R(t)} - \frac{W(t)}{2} dN(t) \tag{1}$$

with $R(t) = q(t)/C + T_p$ where T_p is the propagation delay, $q(t)$ is the queue length at the router, C is the router's transmission capacity, thus $q(t)/C$ is the queuing delay and $R(t)$ is the round trip time delay, and $dN(t)$ is the number of marks the flow suffers. In a network topology of N homogeneous TCP sources and one router a model relating the average value of these variables and the router's queue dynamics becomes[2]

$$\dot{W}(t) = \frac{1}{R(t)} - \frac{W(t)}{2} \frac{W(t - R(t))}{R(t - R(t))} p(t - R(t)) \tag{2}$$

$$\dot{q}(t) = \left[\frac{N(t)}{R(t)} W(t) - C \right]^+ \tag{3}$$

where $p(t)$ is the probability of packet mark due to the ECN mechanism at the router.

In our scheme the dynamics of the new TCP are

$$\dot{W}(t) = \frac{1}{R(t)} - \frac{W(t)}{\beta_1} \frac{W(t - R(t))}{R(t - R(t))} Prob_1(t - R(t)) - \frac{W(t)}{\beta_2} \frac{W(t - R(t))}{R(t - R(t))} Prob_2(t - R(t)) \tag{4}$$

$$\dot{q}(t) = \left[\frac{N(t)}{R(t)} W(t) - C \right]^+ \tag{5}$$

where $Prob_1$ is the probability of receiving a mark '01' and $Prob_2$ is the probability of receiving a mark '11'. Note that $Prob_2 = p_2$ and $Prob_1 = p_1(1 - p_2) \approx p_1$.

Using similar techniques like the ones used in[2] linear models of (2), (3), (4), (5) can be derived, and a control theoretic analysis of these models leads to the following conclusions. In the MECN scheme, there are two regions of operations in the congestion avoidance mode: below and above mid_{th}. We define the D.C. gain of the ECN and MECN scheme respectively as (details of the derivation can be found in a subsequent publication)

$$K_0 = L_{\text{RED}} \frac{(R_0 C)^3}{(2N)^2} \tag{6}$$

$$K_{\text{MECN}} = \frac{(R_0 C)^3}{2N^2} \left(\frac{L_{RED_1}}{\beta_1} + \frac{L_{RED_2}}{\beta_2} \right) \tag{7}$$

where $L_{\text{RED}} = L_{\text{RED}_1} = P_{\text{max1}}/(\text{max}_{\text{th}} - \text{min}_{\text{th}})$ and
$L_{\text{RED}_2} = P_{\text{max2}}/(\text{max}_{\text{th}} - \text{mid}_{\text{th}})$ as shown in Figure 3. R_0 is the equilibrium round trip time and P_{max1} and P_{max2} are the maximum probabilities of packet drops in both the regions. In our case, we have $P_{\text{max1}} = P_{\text{max2}} = P_{\text{max}}$.

From the above, we now proceed to study the effect of Delay Margin and Sensitivity in the ECN and MECN schemes. The Delay Margin is how much additional delay a system can tolerate before going to unstable regions. Hence it is measure of stability. Sensitivity is a measure of how aggressive the system is and the error in the system. We ideally want to work in regions of high Delay Margins and low Sensitivity. We can see from Figure 3, that there are two regions of operation in the congestion avoidance mode. We have one region of operation where the sources go down by β_1 and another region of operation where the sources go down by β_1 and β_2. We proceed to derive the Delay Margin in the MECN system when the system settles in both the regions of congestion avoidance. In this case the Delay Margin is calculated as follows. We define $\varpi_{\text{mecn}} = K\sqrt{(K_{\text{MECN}}^2 - 1)}$. The phase margin of the system is then given by $PM = \pi - \text{Tan}^{-1}(\frac{\varpi_{\text{mecn}}}{K})$. Thus the Delay Margin of the system is given by

$$DM = \frac{PM}{\varpi_{\text{mecn}}} - R_0. \tag{8}$$

Similarly for ECN, we define $\varpi_{\text{ecn}} = K\sqrt{(K_0^2 - 1)}$. The phase margin of the system is then given by $PM = \pi - \text{Tan}^{-1}(\frac{\varpi_{\text{ecn}}}{K})$. Thus the Delay Margin of the system is given by

$$DM = \frac{PM}{\varpi_{\text{ecn}}} - R_0. \tag{9}$$

K is defined as [10]

$$K = -Clog(1 - \alpha) \tag{10}$$

where C is the link capacity of the router and α is the queue averaging parameter.

The ratio of the inverse of the sensitivities of the MECN and ECN scheme is given by $\frac{1+K_{MECN}}{1+K_0}$. If we have this ratio greater than one, we then are more aggressive and have better tracking of the steady state queue. Thus, if the average queue settles below mid_{th}, we have $L_{\text{RED2}} = 0$. Thus we can see that the Delay Margin is higher in the MECN case than the ECN case by comparison of equations 8 and 9. Therefore, in the Multi-Level scheme the queue oscillates less and goes to zero less often resulting in throughput improvement. If however the queue settles above the mid_{th}, we could still have a throughput improvement because in this region, we are interested more in less sensitivity. We have a less sensitivity in MECN scheme from equations 6 and 7. This means that the queue better tracks the steady state queue. Also since we always want to operate in the low delay region, the MECN scheme is more aggressive if the average queue settles above mid_{th} and will push the system to the low delay region faster. It is important to note that we do not want a very high K_{MECN} as in that case we may be compromising on the Delay Margin and so the queue may oscillate more resulting in packet drops and hence reduction in throughput. Thus we have an inherent tradeoff that is shown in Figure 4. We can see that as we move towards the right, we are operating in a region of decreased sensitivity with corresponding decrease in Delay Margin

We thus ideally want to operate in a region where K_{MECN} is high and also ensuring sufficient Delay Margin.

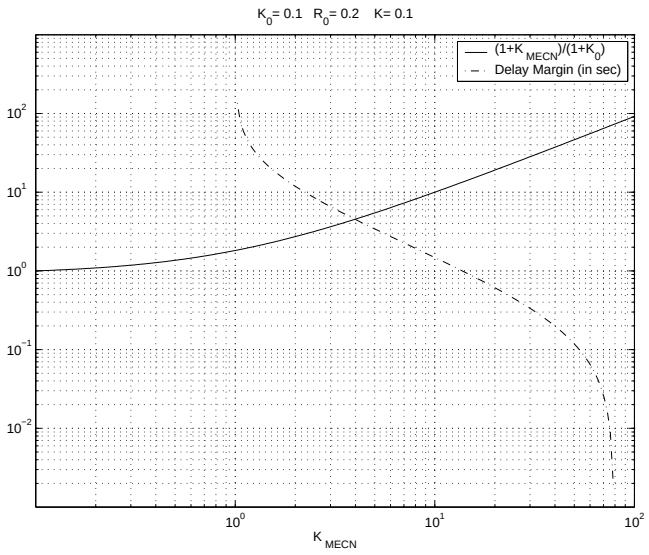


Figure 4. Trade-off between Delay Margin and Sensitivity

4. NS SIMULATIONS

We now use the above results to show that MECN performs significantly better than ECN over a wide range of thresholds.

4.1. Simulation Configuration

For all our simulations we used the following configuration shown in Figure 5. A Number of sources S_1, S_2, S_3,..., S_n are connected to a router R_1 through 10Mbps, 2ms delay links. Router R_1 is connected to R_2 through a 1.5Mbps, 40ms delay link and a number of destinations D_1, D_2, D_3,..., D_n are connected to the router R_2 via 10Mbps 4ms delay links. The link speeds are chosen so that congestion will happen only between routers R_1 and R_2 where our scheme is tested. 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. The number of sources is varied to alter the congestion level. The weight used for queue averaging is $\alpha = 0.002$.

4.2. Results obtained through NS

We first use low threshold levels. For analyzing ECN, we set $\min_{\text{th}}$ as 1 and $\max_{\text{th}}$ as 4 packets. For analyzing MECN, we set $\min_{\text{th}}$ as 1, mid_{th} as 2 and $\max_{\text{th}}$ as 4 packets. Figure 6 shows the instantaneous and average queue of a single level ECN where the oscillations in the queue are very high (the queue goes to zero often). This results in a substantial reduction in the throughput of the router. However the MECN scheme shown in Figure 7 gives a higher throughput as the oscillations are reduced. This implies that the utilization of the router is improved. The control inference of this observation is the fact that the queue is more stable initially and has lesser error in MECN compared to ECN, thus improving performance. Figure 8 compares the link efficiency of ECN with MECN scheme for low thresholds. From Figures 6 and 7 we observe that MECN should give a better Link Efficiency than ECN. This is clearly shown in Figure 8.

In Figures 9 and 10, we compare the performances over a broader range of thresholds. We set $\min_{\text{th}}$ as 30, mid_{th} as 60 and $\max_{\text{th}}$ as 90 packets. An increase in throughput beyond a certain $\max_{\text{th}}$ is not possible. However the MECN scheme outweighs the ECN scheme in that the error in the queue and jitter are reduced.

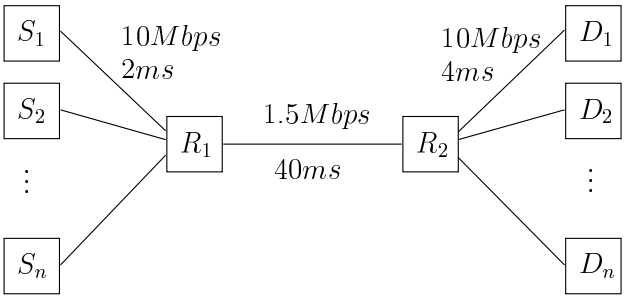


Figure 5. Network configuration for `ns` simulations

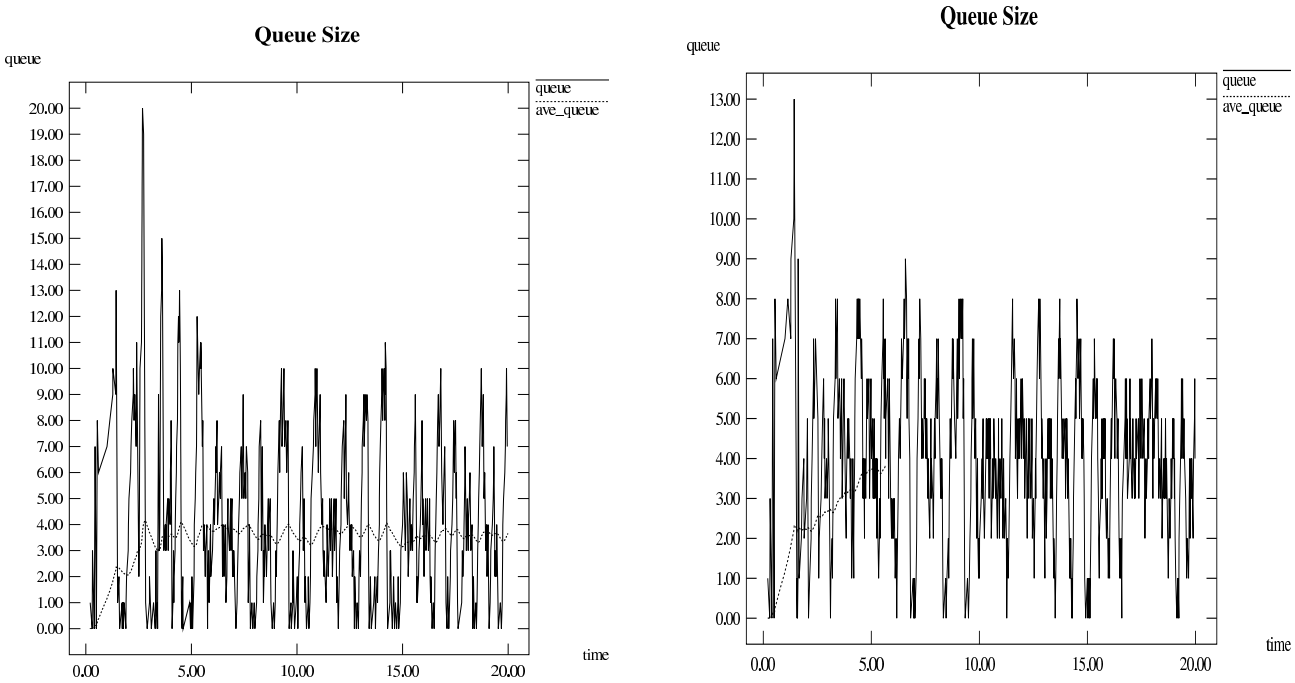


Figure 6. Queue size of ECN for lower delay

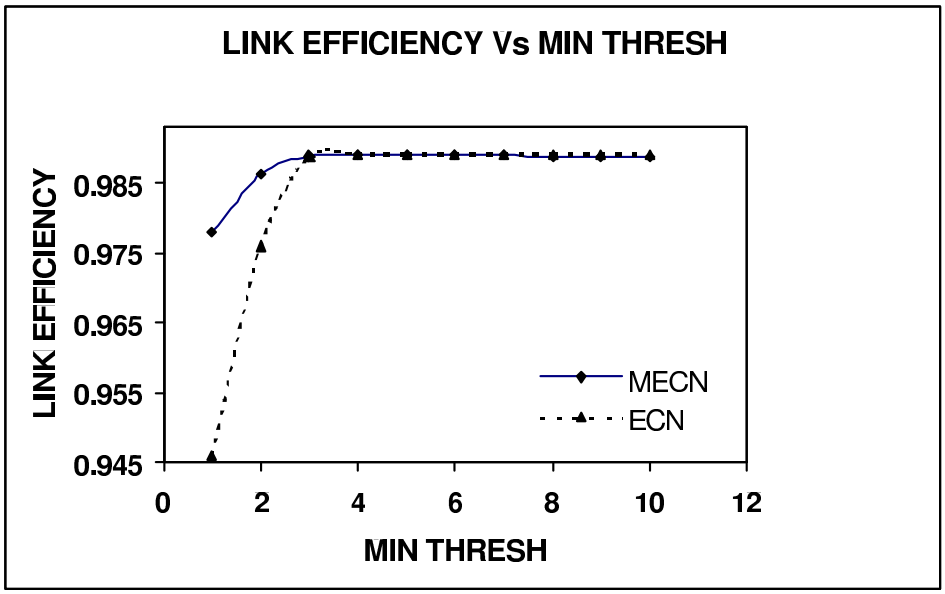


Figure 8. Comparison of link efficiency for ECN and MECN

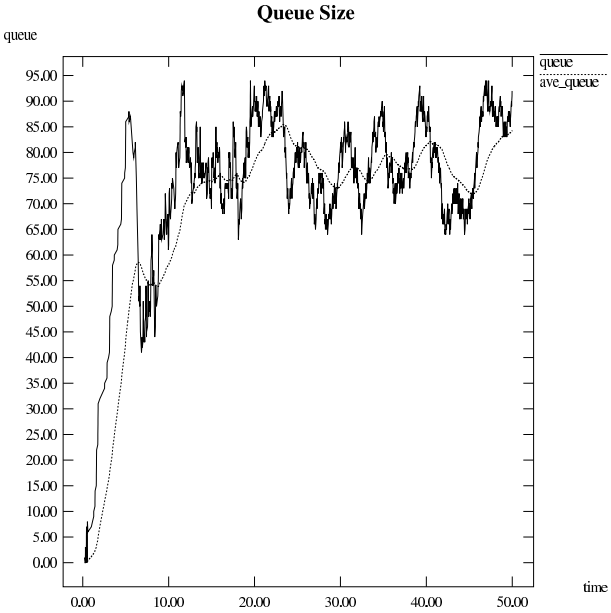


Figure 9. Queue size of ECN for higher delay

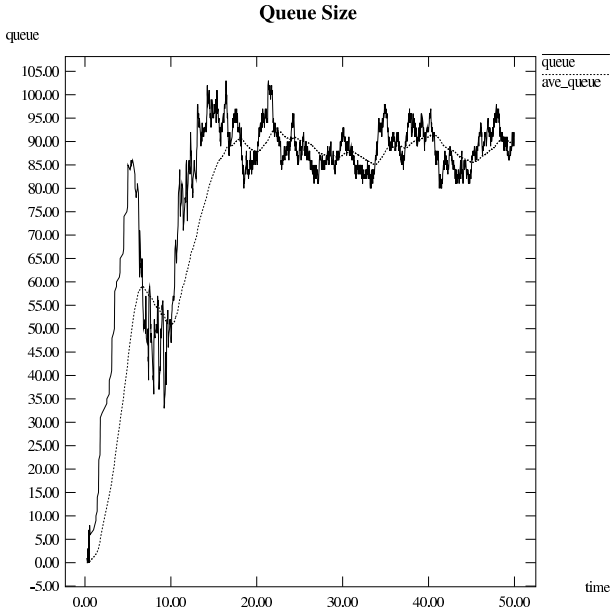


Figure 10. Queue size of MECN for higher delay

Figure 14 shows a plot of Throughput Vs. Average Delay, which is a popular metric for analyzing performance of a router. We can clearly see the improvement in performance of MECN over single level ECN. It is a well known fact that throughput is a trade-off with average delay: we want higher throughput with lesser delay. From the figure we can see that MECN gives the same throughput as single level ECN with a lesser average delay. MECN* indicates fine tuned MECN (larger K_{MECN}) using the performance improvement equation derived in Section 3. As we can see better performance can be obtained by having a higher K_{MECN}.

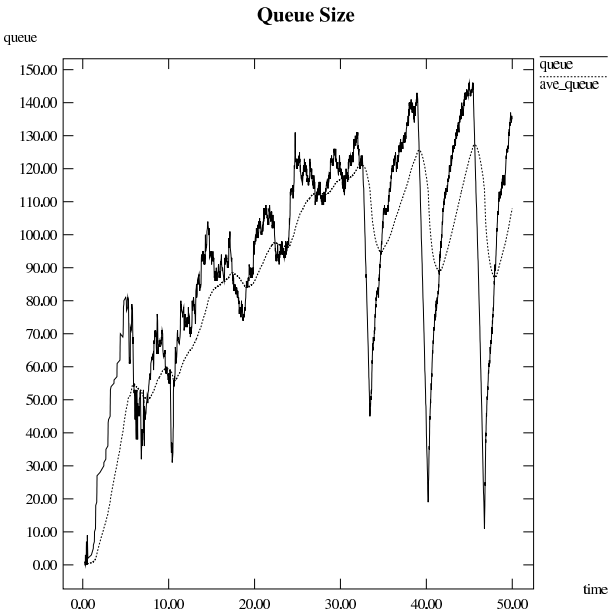


Figure 11. Effect of parameter tuning on MECN: low K_{MECN}

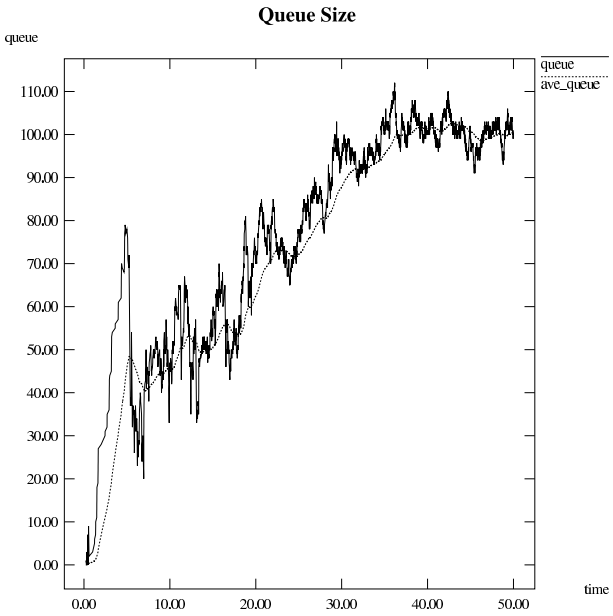


Figure 12. Effect of parameter tuning on MECN: high K_{MECN}

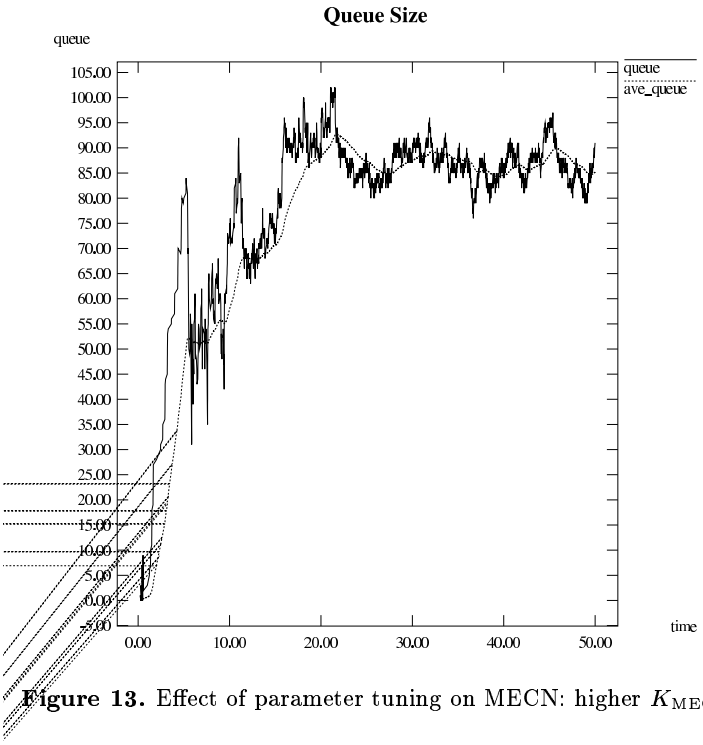


Figure 13. Effect of parameter tuning on MECN: higher K_{MECN}

5. CONCLUSIONS

In this paper, we have studied the performance of a Multi-Level Explicit Congestion Notification scheme. Two bits are now used to indicate four levels of congestion. We change the marking algorithm of RED and the TCP

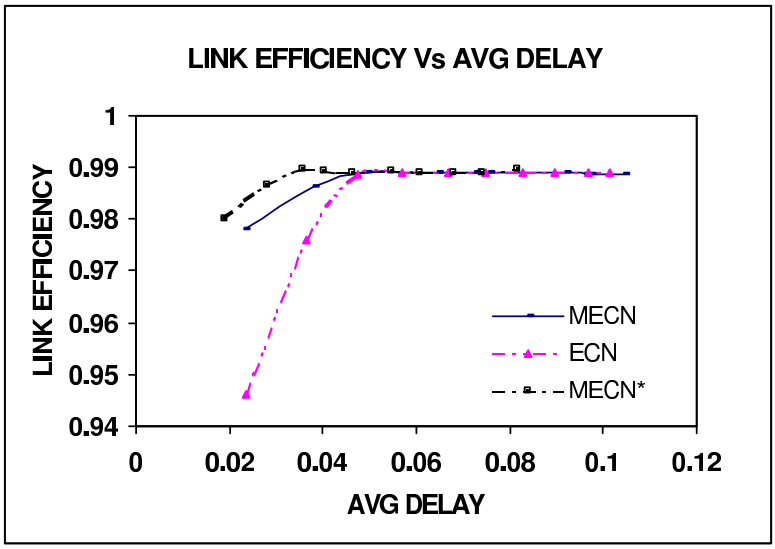


Figure 14. Comparison of link efficiency Vs average delay for ECN and MECN

source reaction to it. We have explained the performance improvement of MECN over ECN using classical control theory and have validated our results using `ns` simulations. For low thresholds, we get a much higher throughput from the router with lesser delays in the case of MECN. For higher thresholds, the improvement is seen in the reduction in the jitter experienced by the flows. We have also provided guidelines for tuning the parameters for optimal performance. We are currently working on robust AQM schemes that would guarantee stable operation with current TCP for uncertain network parameters, thus needing less tuning than existing schemes.

ACKNOWLEDGMENTS

This research was supported in part by NSF grants CISE-9980637, ANI-0073725 and by a grant from OAI, Cleveland, Ohio.

REFERENCES

1. A. Durresi, M. Sridharan, C. Liu, M. Goyal, and R. Jain, "Multilevel explicit congestion notification," in *Proc. of the 5th World Multiconference on Systemics, Cybernetics and Informatics SCI'2001, ABR over the Internet*, **12**, pp. 12–16, (Orlando, FL), July 22–25 2001.
2. C. V. Hollot, V. Misra, D. Towsley, and W. B. Gong, "Analysis and design of controllers for AQM routers supporting TCP flows," `http://www.cs.columbia.edu/~misra/pubs/TAC_special.pdf` , 2002.
3. S. H. Low, F. Paganini, and J. C. Doyle, "Internet congestion control," *IEEE Control Systems Magazine* **22**, pp. 28–43, 2002.
4. W. Feng, D. Kandlur, D. Saha, and K. Shin, "A self-configuring RED gateway," in *Proc. of INFOCOM*, pp. 1320–1328, March 1999.
5. S. Floyd, R. Gummadi, and S. Shenker, "Adaptive RED, an algorithm for increasing the robustness of RED's active queue management." unpublished, 2001.
6. D. Bansal, H. Balakrishnan, S. Floyd, and S. Shenker, "Dynamic behavior of slowly responsive congestion control algorithms," in *Proc. of SIGCOMM*, 2001.
7. V. Jacobson, "Congestion avoidance and control," in *Proc. of ACM/SIGCOMM*, pp. 314–329, August 1988.
8. S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking* **1**, pp. 397–413, August 1993.
9. K. Ramakrishnan and S. Floyd, "A proposal to add explicit congestion notification (ECN) to IP." RFC 2481, January 1999.
10. V. Misra, W. B. Gong, and D. Towsley, "Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED," in *Proc. of ACM/SIGCOMM*, pp. 151–160, 2000.
11. S. Floyd, "TCP and explicit congestion notification," *ACM Computer Communication Review* **24**, pp. 10–23, October 1994.
12. H. Özbay, *Introduction to Feedback Control Theory*, CRC Press LCC, Boca Raton FL, 1999.