Control Theory Optimization of MECN in Satellite Networks

Arjan Durresi
Louisiana State University, USA
Department of Computer Science
durresi@csc.lsu.edu

Mukundan Sridharan, Sriram Chellappan,
Raj Jain, Hitai Özbai
{

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

Leonard Barolli
Fukuoka Institute of Technology, Japan
barolli@fit.ac.jp

Abstract

Congestion in the Internet results in wasted bandwidth and also stands in the way of guaranteeing QoS. The effect of congestion is multiplied many fold in Satellite networks, where the resources are very expensive. Thus congestion control has a special significance in the performance of Satellite networks. In today's Internet, congestion control is implemented mostly using some of the de facto standard, RED. We have proposed previously a new scheme called the Multi-level Explicit Congestion Notification (MECN), which works with the framework of ECN, but enhances the performance using multiple level congestion feedback from the network. But like RED, the tuning of MECN could be a problem. Tuning of parameters is necessary for achieving high throughput with corresponding low delays. Also it is desirable to keep the oscillations in the queue low to reduce jitter, which is the major concern in real-time applications such as voice or video over IP. In this paper, we use a linearized fluid flow model of TCP-MECN to study the performance and stability of the Queue in the router. We use classical control tools such as Delay Margin and Tracking Error to provide guidelines for optimizing satellite IP networks, which use MECN. We apply our results for optimizing the performance of satellite networks. We use ns simulator to validate the results of our analysis.

1. Introduction

The rapid globalization of the telecommunications industry and the exponential growth of the Internet are increasing the demands for ubiquitous network access. Satellite networks can be an integral part of the newly emerging global information infrastructures [2]. Satellite networks offer global coverage, broadcast capabilities, flexibility in bandwidth allocation, support for mobility and short time for implementing services, even in areas with little infrastructure. All these qualities make satellite networks a good candidate for the future Internet infrastructure for broadband access network, high-speed backbone and combination of both of them. Therefore, in recent years, significant investments have been made in the planning and development of broadband satellite networks. However, to meet this challenges, satellite networks should provide Qualify-of-Service (QoS). Congestion remains the major obstacle to Quality of Service (QoS) on the Internet. Congestion is a critical problem especially in satellite networks, where TCP congestion control performance is affected by intrinsic satellite link characteristics such as latency, packet loss due to congestion and losses due to transmission errors links [3]. Although a number of schemes have been proposed for network congestion control in satellite networks, the search for new schemes continues [4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15, 16, 17, 18]. Among these schemes, the most used is RED [5], which is part of Active Queue Management (AQM) class of algorithms.

RED and a slightly different algorithm Early Congestion Notification (ECN) are much more powerful than the simple packet drop indication used by existing routers and is more suitable for high distance-bandwidth networks. We showed in [1], that a enhanced version of ECN, called the Multi-level explicit congestion notification (MECN), performs better than ECN. As pointed out in [19] one of RED's main weakness is that the average queue size varies with the level of congestion and with parameter settings. As a result the average queuing delay in RED is sensitive to the traffic load and to parameters. Unfortunately, MECN also suffers from the above mentioned drawbacks. The average queuing delay is particularly very important for QoS applications. So setting the parameters of MECN is very important and also the range of variation of the parameters or

Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'05)
1545-0678/05 $20.00 © 2005 IEEE

the traffic load, for which the delay or the throughput performance doesn't vary significantly, is also important. As the level of traffic in the network keeps changing dynamically, it is important to find out the range of traffic for which given parameter settings remain valid. We analyze here the TCP-MECN behavior using control theory and provide essential guidelines, suing Delay Margin and Tracking Error minimization [20], for setting the parameters and also suggest ways to find a a bound on traffic load and other MECN parameters, so that the delay, throughput and jitter performance doesn't degrade appreciably. The rest of the paper is organized as follows. In Section 2, we give an introduction of MECN protocol. In Section 3, we analyze stability and performance TCP-MECN using the fluid-flow model. In Section 4, we use simulations to give guidelines based on the analysis in section 3. We also validate the results obtained using control theory using ns simulator [21]. In Section 5, we describe the network configuration used for the simulations. In Section 7, we present the conclusions of our research.

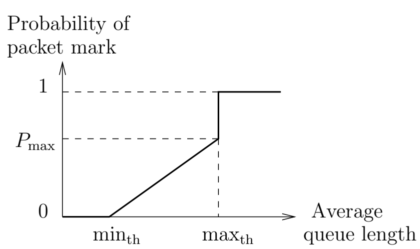

Figure 1. Probabilities of marking packets for the new scheme

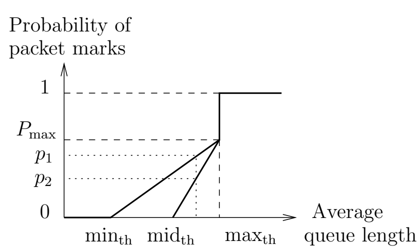

Figure 2. Probabilities of marking packets for the new scheme

2. Multi-level Explicit Congestion Notification (MECN)

2.1. Marking 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 four 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 $\mathrm{mid}_{\rm th}$, in addition to the $\min_{\rm th}$ and $\max_{\rm th}$. If the size of the average queue is in between $\min_{\rm th}$ and $\min_{\rm th}$, 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 $\mathrm{mid}_{\rm th}$ and $\max_{\rm th}$, 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 $\max_{\rm th}$ all packets are dropped. The packet dropping policy of RED is shown in Figure 1. The modified packet marking/dropping policy of MECN is shown in Figure 2.

Table 1. Router Response to Congestion: Marking of CE and ECT bits and packet dropping

CEbit	ECTbit	Congestion State
0	1	No Congestion
1	0	Incipeint Congestion
1	1	Moderate Congestion
Packet	Drop	Severe Congestion

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 two-level marking in the traditional ECN, we make use of three 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. And in piggybacked acknowledgements, '10' and '11' indicated noncongestion and congestion, with the receiver source indicating that the congestion window has been reduced. In our scheme, if the source has to indicate that the congestion window has been reduced, then the congestion information has to wait for the next packet. In this case the congestion information is ignored. But this will not cause any major problems to the scheme because, if the congestion is persistent then a lot of packets are going to get marked and the received source will eventually get the congestion information. So in the new scheme, '00' will indicate congestion window reduced, '01' will indicate no congestion, '10' will indicate mild congestion and '11' will indicate heavy congestion. The packet drop is recognized using traditional ways, by timeouts or duplicate ACKs. The marking in the ACKs CWR, ECE bits is shown in Table 2.

Table 2. End Host reflecting Congestion Information: Marking of CWR and ECE bits

CWRbit	ECEbit	Congestion State
0	0	Congestion Window Reduced
0	1	No Congestion
1	0	Incipeint Congestion
1	1	Moderate Congestion

2.3. Response of TCP source

We believe that the marking of ECN should not be treated in the same way as a packet drop, because 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 multiplicatively by $\beta_3 = 50\%$. 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 do not 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, the *cwnd* is allowed to grow additively as usual. When the marking is '10' (incipient congestion), *cwnd* is decreased by β_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 β_2% less than 50% but more than 1. In Table 3 are shown the TCP source responses and the value of βs we have implemented. Another method could be to decrease additively the window, when the marking is '10' (incipient congestion), instead of maintaining the window. This will be analyzed in future study.

Table 3. TCP Source response

Congestion State	CWND Change
No Congestion	Increase additively
Incipient Congestion	Decreace by $\beta_1 = 20\%$
Moderate Congestion	Decreace by $\beta_2 = 40\%$
Severe Congestion	Decreace by $\beta_3 = 50\%$

If the average queue length is less than $\mathrm{mid_{th}}$, then the modified-TCP congestion windows corresponding to the marks '10' keep increasing by 1 every round-trip time in congestion avoidance mode, thus linearly increasing the sending rates of these flows. Consequently, the average queue length will keep increasing unless some marks '11' are received by the sources, which correspond to operating in the region where the average queue length is larger than $\mathrm{mid_{th}}$. We can thus conclude that the steady-state average queue length is larger than $\mathrm{mid_{th}}$.

3. Stability And Performance Analysis Using the Mathematical Model

In the following we do the linearization of the new system. In effect, we derive the transfer functions when the average queue length is between $\mathrm{mid_{th}}$ and $\mathrm{max_{th}}$. 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 our scheme the dynamics of the new TCP are derived from [14]

$$\begin{aligned}\dot{W}(t) &= \frac{1}{R(t)} - \frac{W(t)}{\beta_1}\frac{W(t-R(t))}{R(t-R(t))}Prob_1(t-R(t)) \\ &\quad - \frac{W(t)}{\beta_2}\frac{W(t-R(t))}{R(t-R(t))}Prob_2(t-R(t)) \qquad (1)\end{aligned}$$

$$\dot{q}(t) = \begin{cases} \frac{N(t)}{R(t)}W(t) - C & \text{if } q(t) > 0 \\ \max\left\{0, \frac{N(t)}{R(t)}W(t) - C\right\} & \text{if } q(t) = 0 \end{cases} \qquad (2)$$

where $Prob_1$ is the probability of receiving a mark '01' and $Prob_2$ is the probability of receiving a mark '11', thus

1545-0678/05 $20.00 © 2005 IEEE

$Prob_2 = p_2$ and $Prob_1 = p_1(1-p_2)$. Using similar techniques than the ones used in [14] a linear model of (1) and (2) can be derived. We first assume that the number of TCP flows and the outgoing link capacity are constant. The operating point $(W_0, q_0, R_0, p_{1_0}, p_{2_0})$ defined by $\dot{W}(t) = 0$ and $\dot{q}(t) = 0$ satisfies

$$W_0^2\left(\frac{p_{1_0}}{\beta_1}(1-p_{2_0}) + \frac{p_{2_0}}{\beta_2}\right) = 1 \tag{3}$$

$$p_{1_0} = (q_0 - \mathrm{min_{th}})L_{RED_1} \tag{4}$$

$$p_{2_0} = (q_0 - \mathrm{mid_{th}})L_{RED_2} \tag{5}$$

$$\tag{6}$$

$$W_0 = \frac{R_0 C}{N} \tag{7}$$

$$R_0 = \frac{q_0}{C} + T_p \tag{8}$$

with $L_{RED_1} = P_{\max}/(\mathrm{max_{th}} - \mathrm{min_{th}})$, $L_{RED_2} = P_{\max}/(\mathrm{max_{th}} - \mathrm{mid_{th}})$ as shown in Figure 2. Next the time-varying nature of the round-trip time delay in the terms "$t - R(t)$" is ignored and these terms are approximated by "$t - R_0$". However the queue length still depends on the round-trip time in the dynamic equation (2).

Let's define

$$f(W, W_R, q, q_R, p_{1_R}, p_{2_R}) = \frac{1}{\frac{q}{C} + T_p} - \frac{W W_R}{\frac{q_R}{C} + T_p}\left(\frac{p_{1_R}}{\beta_1}(1-p_{2_R}) + \frac{p_{2_R}}{\beta_2}\right) \tag{9}$$

$$g(W, q) = \frac{N}{\frac{q_R}{C} + T_p} W - C \tag{10}$$

where $W_R(t) = W(t - R_0)$, $q_R(t) = q(t - R_0)$, $p_{1_R}(t) = p_1(t - R_0)$ and $p_{2_R}(t) = p_2(t - R_0)$. By evaluating partials of f and g at the operating point and using similar techniques to the ones used in [14], we obtain the linearized transfer function as and neglecting some high-frequency dynamics similarly to what was done in [14], the linearized dynamics of TCP-MECN results in the open-loop transfer function

$$G(s) = \frac{1}{\left(\frac{R_0^2 C}{2N}s + 1\right)(R_0 s + 1)} \cdot K_{\mathrm{MECN}} \cdot \frac{\mathrm{e}^{-R_0 s}}{\frac{s}{K} + 1} \tag{11}$$

with

$$K_{\mathrm{MECN}} = \frac{R_0^3 C^3}{2N^2}\left[\frac{1-p_{2_0}}{\beta_1}L_{RED_1} + \left(\frac{1}{\beta_2} - \frac{p_{1_0}}{\beta_1}\right)L_{RED_2}\right]. \tag{12}$$

where $L_{RED_1} = P_{\max}/(\mathrm{max_{th}} - \mathrm{min_{th}})$, $L_{RED_2} = P_{\max}/(\mathrm{max_{th}} - \mathrm{mid_{th}})$ as shown in Figure 2. p_{1_0} and p_{2_0} are solutions of (6) with

$$p_{1_0} = \frac{P_{\max}}{\mathrm{max_{th}} - \mathrm{min_{th}}}(q_0 - \mathrm{min_{th}}) \tag{13}$$

$$p_{2_0} = \frac{P_{\max}}{\mathrm{max_{th}} - \mathrm{mid_{th}}}(q_0 - \mathrm{mid_{th}}). \tag{14}$$

Similarly to [14] we assume that the low-pass filter pole K is less than the corner frequencies of the new TCP, and that it dominates the closed-loop system behavior. The unity-gain crossover frequency ω_g (i.e. $|G(j\omega_g)| = 1$) thus satisfies

$$\omega_g \ll \min\left\{\frac{2N}{R_0^2 C}, \frac{1}{R_0}\right\}. \tag{15}$$

Then at low frequency we have

$$G_0(s) \approx \mathrm{e}^{-R_0 s}\frac{K_0}{\frac{s}{K} + 1} \tag{16}$$

and

$$G(s) \approx \mathrm{e}^{-R_0 s}\frac{K_{\mathrm{MECN}}}{\frac{s}{K} + 1}. \tag{17}$$

3.1. Stability analysis

The Delay Margin is also a parameter of interest here. The Delay Margin is a measure of the stability of the system (low oscillations). The Phase Margins of the systems without delay are (see for example [20])

$$PM(\omega_g) \approx \pi - \tan^{-1}\left(\frac{\omega_g}{K}\right) \tag{18}$$

where ω_g is such that $|G(j\omega_g)| = 1$, i.e. $\omega_{g_0} = K\sqrt{K_0^2 - 1}$ for the traditional TCP-ECN, and $\omega_g = K\sqrt{K_{\mathrm{MECN}}^2 - 1}$ for TCP-MECN. The Delay Margin (DM), which represents how much the round-trip time can be increased without violating stability of the feedback system (see for example [20]), is then:

$$DM(\omega_g) \approx \frac{PM(\omega_g)}{\omega_g} - R_0 \tag{19}$$

$$\approx \frac{\pi - \tan^{-1}\left(\frac{\omega_g}{K}\right)}{\omega_g} - R_0. \tag{20}$$

If $K_{\mathrm{MECN}} > K_0$ we have $\omega_g > \omega_{g_0}$, and since $DM(\omega_g)$ is a decreasing function of ω_g we have a decrease in the Delay Margin in using TCP-MECN. The reason for studying stability in this regions is because queue oscillations around here can lead to packets being dropped if the queue crosses the $\mathrm{max_{th}}$. But this is the price we pay for having an improved performance at low frequency while still using such simple feedback control mechanism. Increasing K_{MECN} further will mean more oscillations which will lead to packet drops.

3.2. Tracking Error Analysis

The steady state *Tracking Error* is the amount of error between the instantaneous queue and the steady state queue.

1545-0678/05 $20.00 © 2005 **IEEE**

The *Tracking Error* refers to how good the system tracks the desired steady state queue and oscillations around it. In networks, the variation in the delays (jitter) is also an important parameter. This is related to better tracking of the steady state queue. For this purpose also we study the *Tracking Error* in the system. Ideally we would like the *Tracking Error* to be small as the system then tracks the steady state queue better. For a fixed propagation delay, it is desirable to operate in a region where we have low *Tracking Error* and high *Delay Margin*. Such a system will guarantee better stability and performance from the point of throughput and jitter.

For the above system the steady state Tracking Error $\mathrm{e_{ss}}$ is defined by [20],

$$\mathrm{e_{ss}} = \lim s \rightarrow 0 s * E(s) \tag{21}$$

where $E(s)$ is the error to a step input in the system given by

$$E(s) = \frac{1}{1+G(s)} * \frac{1}{s} \tag{22}$$

The steady state error is then given by

$$\mathrm{e_{ss}} = \frac{1}{1+G(0)} \tag{23}$$

where $G(0) = K_{\mathrm{MECN}}$

4. Tuning Effects and Guidelines for Parameter Setting

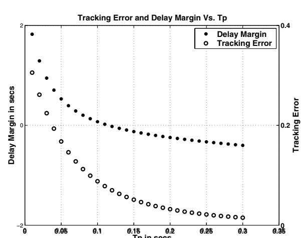

Figure 3. Tracking Error And Delay Margin for unstable GEO Satellite Network

We analyze the performance of GEO satellite networks with the following parameters. One-way latency, $\mathrm{T_p}$ = 250ms, $\mathrm{max_{th}}$= 60 packets, $\mathrm{min_{th}}$= 20 packets, C= 250 packets/second, $\mathrm{P_{max}}$= 0.1, N = 5, α = 0.002. The network configuration is shown in Figure 9.

For the above case, we plot the *Tracking Error* and the *Delay Margin* as a function of the propagation time *Tp* in

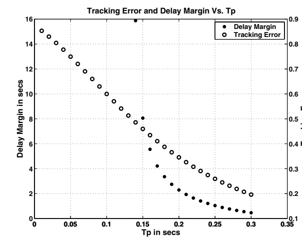

Figure 4. Tracking Error And Delay Margin for stable GEO Satellite Network

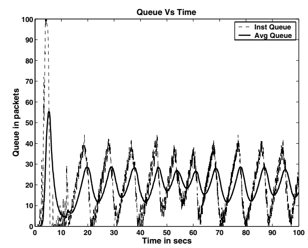

Figure 5. Queue for an unstable GEO Satellite Network

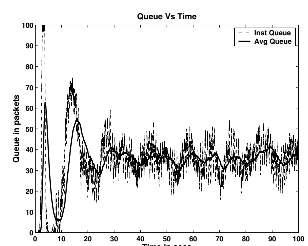

Figure 6. Queue for a stable GEO Satellite Network

Figure 7. Jitter vs. Tracking Error for a GEO Satellite Network

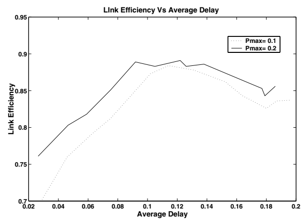

Figure 8. Comparison of Throughput for two different G(0) in GEO Satellite Networks

Figure 3. It can be seen that the system has a negative Delay Margin, which means that the system is unstable. Since we want to operate in regions where the queuing delay will be less, we are concerned about the oscillations in the queue as if they go to zero, there will be reduction is throughput, since whenever the queue goes to zero the link is under utilized. The ns simulation result for this case is shown in Figure 5. The ns simulation configuration details are given in Section 5. From Figure 5, we can see the high oscillations in the queue. Since the queue goes to zero often, there is less throughput of the system.

We now try to improve the performance of the above system. We want the system to have a positive *Delay Margin*. We thus need to decrease the gain, K_{MECN} of the system from equation 3. We do this by increasing the *N* to 30. The *Delay Margin* in this case is positive (approximately 1.0), as show in Figure 4 and the system is expected to oscillate much less. This reduction is oscillation makes the queue to go to zero less often (actually never), and gives an increase in throughput, in low delay regions. The ns results for this case is shown in Figure 8.

For the given load level *N*, we obtain a maximum allowable $\mathrm{P_{max}}$ that guarantees a positive *Delay Margin* for system parameters $\mathrm{max_{th}}$= 4 packets, $\mathrm{min_{th}}$= 1 packets, C= 250 packets/second, N = 30, α = 0.002. The maximum value of $\mathrm{P_{max}}$ calculated from equation 20 that gives a positive *Delay Margin* is 0.3. Thus the system is stable for any $\mathrm{P_{max}}$ less than 0.3. Having this result we now proceed to tune the system for better performance. Our main goals are stability with minimum *Tracking Error*. We now test our system by changing K_{MECN}, such that the system remains in stable region. A high K_{MECN} system means a system with reduced *Tracking Error* from equation 3.2. Such a system will give better throughput performance and lower jitter. In Figure 7 is shown the dependance of jitter from *Tracking Error*.

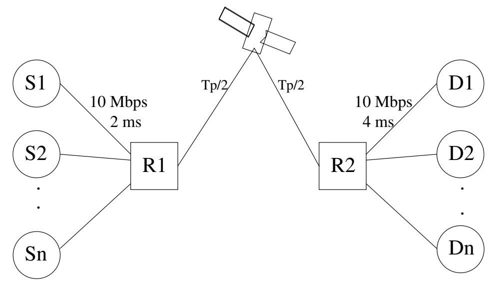

Figure 9. Satellite Network Configuration

5. NS Simulation Configuration

For all our simulations we used the following configuration, shown in Figure 9. A number of sources $S_1, S_2, \ldots, S_n$ are connected to a router R_1 through 10Mbps, wms delay links. Router R_1 is connected to the Satellite Router through a 2Mbps, $\mathrm{T_p}$ ms delay link. The Satellite Router is connected to R_2 through a 2Mbps, $\mathrm{T_p}$ ms delay link and a number of destinations $D_1, D_1, \ldots, D_n$ are connected to the router R_2 via 10Mbps, 4ms delay links. The link speeds are chosen so that the congestion will happen only between routers R_1 and the Satellite Router, where our scheme is tested.

This configuration can simulate the case of different type of satellites by varying the delay $\mathrm{T_p}$. A delay of 250ms is used for $\mathrm{T_p}$ GEO satellites. An FTP application runs on each source. Reno TCP is used as the transport agent. 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 α = 0.002.

6. Acknowledgements

This research was supported in part by the NSF grant # 0413187.

7. Conclusions and Future Work

In this paper, we studied the performance of a Multi-Level Explicit Congestion Notification scheme. Two bits are now used to indicate four levels of congestion. The marking algorithm of RED and the TCP source reaction to it are changed. We have explained the performance improvements of MECN over ECN using classical control theory tools and these results have been validated by *ns* simulations. For low thresholds, we get a much higher throughput from the router with lesser delays using MECN compared to ECN. For higher thresholds, the improvement is seen in the reduction in the jitter experienced by the flows.

The Multi-level marking architecture and be extended to several other schemes, which now use just single level marking (like several variants of RED). We are currently studying the effects of Multi-level marking on several other RED based marking schemes and load based schemes.

References

[1] A. Durresi, M. Sridharan, C. Liu, M. Goyal, and R. jain, "Traffic management using multilevel explicit congestion notification," in *Proc. of the 5th World MultiConference on Systemics, Cybernetics and Informatics SCI'2001, ABR over the Internet*, (Orlando, FL), pp. 12–17, July 22-25 2001.

[2] A. Durresi, S. Kota, R. jain, and M. Goyal, "Achieving qos for tcp traffic in satellite networks with differentiated services," in *Proc. of the Space Communications*, vol. 17, pp. 125–136, 2001.

[3] S. Kota, "Multimedia satellite networks: Issues and challenges," *Proc. SPIE International Symposium on Voice,Video and Data Communications,Boston*, Nov 1-5 1998.

[4] K. Ramakrishnan and S. Floyd, "A proposal to add explicit congestion notification (ECN) to IP." RFC 2481, January 1999.

[5] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 397–413, August 1993.

[6] D. D. Clark and W. Fang, "Explicit allocation of best-effort packet delivery service," *IEEE Trans. on Networking*, vol. 6, pp. 362–373, August 1998.

[7] W. Feng, D. Kandlur, D. Saha, and K. Shin, "BLUE: a new class of active queue management algorithms." University of Michigan Technical Report CSE-TR-387-99, 1999.

[8] S. Kalyanaraman, R. Jain, S. Fahmy, R. Goyal, and B. Vandalore, "ERICA switch algorithm for ABR traffic management in ATM networks," *IEEE/ACM Transactions on Networking*, vol. 8, no. 1, pp. 87–98, 2000.

[9] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the Internet," *IEEE/ACM Transactions on Networking*, vol. 7, no. 4, pp. 458–472, 1999.

[10] S. Floyd and K. Fall, "Router mechanisms to support end-to-end congestion control," 1997.

[11] J. M. M Mathis, J Semke and T. Ott, "The macroscopic behavious of the tcp congestion avoidance algorithm," *Computer Communications Review*, vol. 27, pp. 67–82, July 1997.

[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, pp. 1–14, July 1989.

[13] S. Floyd and T. Henderson, "The NewReno modification to TCP's fast recovery algorithm." Internet RFC 2582, Experimental, April 1999.

[14] C.V. Hollot, V. Misra, D. Towsley and W.B. Gong, "Analysis and Design of Controllers for AQM Routers Supporting TCP Flows," *IEEE Transactions on Automatic Control*, vol. 47, no. 6, pp. 945-959, 2002

[15] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP selective acknowledgement options." IETF Internet RFC 2018, October 1996.

[16] K. K. Ramakrishnan and R. Jain, "A binary feedback scheme for congestion avoidance in computer networks," *ACM Transcations on Computer Systems*, vol. 8, pp. 158–181, May 1990.

[17] S. K. R Jain and R. Viswanathan, "Rate based schemes:mistakes to avoid," *AF-TM 940882*, September 1994.

[18] S. K. R Jain and R. Viswanathan, "Ordered becn:why we need a timestamp or sequence number in the rm cell," *ATM Forum 940987*, October 1994.

[19] T. Ott, T. Lakshman, and L. Wong, "SRED: Stabilized RED," in *Proc. of IEEE INFOCOM*, 1999.

[20] H. Özbay, *Introduction to Feedback Control Theory*. Boca Raton FL: CRC Press LCC, 1999.

[21] http://www.isi.edu/nsnam/ns/