Tuning RED Parameters in Satellite Networks Using Control Theory

Mukundan Sridharan[a], Arjan Durresi*[a], Sriram Chellappan[b], Hitay Ozbay[b], Raj Jain[a]
[a]Department of Computer And Information Science
[b]Department of Electrical Engineering
The Ohio State University
2015 Neil Avenue, Columbus, OH 43210, USA.
{sridhara, durresi, jain} @ cis.ohio-state.edu, {chellappan.1, ozbay.1} @osu.edu
*phone: 1 614 688 510; fax: 1 614 292 2911

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 form of the de facto standard, RED. But tuning of parameters in RED has been a major problem throughout. Achieving high throughput with corresponding low delays is the main goal in parameter setting. It is also desired to keep the oscillations in the queue low to reduce jitter, so that the QoS guarantees can be improved. In this paper, we use a previously linearized fluid flow model of TCP-RED to study the performance and stability of the Queue in the router. We use classical control tools like Tracking Error minimization and Delay Margin to study the performance, stability of the system. We use the above-mentioned tools to provide guidelines for setting the parameters in RED, such that the throughput, delay and jitter of the system are optimized. Thus we provide guidelines for optimizing satellite IP networks. We apply our results exclusively for optimizing the performance of satellite networks, where the effects of congestion are much pronounced and need for optimization is much important. We use *ns* simulator to validate our results to support our analysis.

Keywords: Satellite Networks, RED, ECN, Fluid-Flow model, QoS, Congestion control, AQM.

1. INTRODUCTION

The rapid globalization of the telecommunications industry and the exponential growth of the Internet are increasing the demands on global telecommunications. Satellite communication networks can be an integral part of the newly emerging national and global information infrastructures [1]. 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 as broadband access network, high-speed backbone and combination of both of them. However, to meet this goal, provisioning of quality-of-service (QoS) within the advanced satellite network systems is the critical requirement. Congestion remains the main 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, bandwidth, packet loss due to congestion, and losses due to transmission errors links [2]. Although a number of schemes have been proposed for network congestion control in satellite networks, the search for new schemes continues [3-13,16,17]. But the winner, at least for the time being seems the class of Active Queue Management (AQM) algorithms, which is one form or other of the RED [6] algorithm.

However, as has been pointed out in [19-21] one of RED's main weaknesses is that the average queue size varies with the level of congestion and with the parameter settings. That is, when the link is lightly congested and/or max_p is high, the average queue size is near min_{th}; when the link is more heavily congested and/or max_p is low, the average queue size is closer to, or even above, max_{th}. As a result, the average queuing delay in RED is sensitive to the traffic load and to parameters. Also RED has a trade off between stability and speed of response. This average queuing delay is very important for QoS applications. So setting the parameters of RED is very important and also the range of variation of parameters or the traffic load, for which the delay or the throughput performance doesn't vary appreciably, is also important. As the level of traffic in the any network keeps changing dynamically, it is very important to find out the range of traffic for which given parameter settings remain valid. We analyze here the TCP-RED behavior using control theory and provide essential guidelines, using Delay Margin and Tracking Error minimization [15], for setting the parameters and also suggest ways to find a bound on traffic load and other RED parameters, so that the delay, throughput and jitter performance doesn't degrade. The rest of the paper is organized as follows. In Section 2, we give the stability and performance analysis of the TCP-RED model, based on the fluid flow model. In Section 3, we use simulations to give guidelines based on the analysis in Section 2. We also validate the results obtained using control theory using ns simulations [14]. In Section 4 we describe the network configuration used to do the simulations. In Section 5, we present the conclusions of our research.

2. STABILITY AND PERFORMANCE ANALYSIS

Figure 1, shows the RED packet drop policy. We use here the already linearized model of TCP-RED scheme derived in [18]. In that, the authors used Bode Gain and Phase margins to analyze the stability of the system. Here the Delay Margin and Tracking Error of the system are the parameters of interest. A stable system might become unstable if the delay in the system is more than a specified limit. So the Delay Margin will give us more insight into the stability of the system especially one with prominent delays in the feedback path. The Delay Margin is the amount of additional delay that can be in the feedback path up to which the system can still be stable [15]. In our case the delay in the feedback is the sum of the queuing delay and the fixed propagation delay. Thus ideally we want the system to have high Delay Margins as that will mean that small variation in delays will still make the system operate in the stable region. If the queue oscillates less in the low delay region, then it will go to zero less often. This is tantamount to an increase in the throughput in such regions. This is because the router can service more packets if the router queue is non-empty. The effect of oscillations in the queue in the low-delay region is thus especially important. In fact in the low-delay region throughput is the main parameter of interest. The Tracking error is also important here. The Tracking error refers to how good the system tracks the desired steady state queue and oscillations around it. In actual 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 decreased Tracking Error and high Delay Margin. Such a system will guarantee better stability and performance from the point of throughput and jitter. We will now proceed to derive expressions for the Delay Margin and Tracking error.

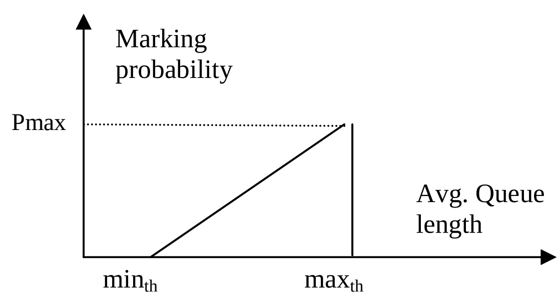

Figure 1. RED packet-drop policy.

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 [18] by the following differential equation

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

In a network topology of N homogeneous TCP sources and one router, an equation for the queue dynamics is [18],

$$\dot{q}(t) = \frac{W(t)}{R(t)} N(t) - C \quad (2)$$

A block diagram describing the TCP dynamics is shown in Figure 2, where C(s) is the AQM controller and P(s) is the TCP plant. The Transfer function for P(s) and C(s) are given below [18].

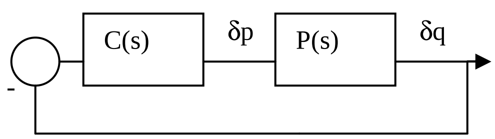

Figure 2. Block Diagram of TCP-RED scheme.

$$P(s) = \frac{\left(\frac{C^2}{2N}\right) e^{-sR_o}}{\left(s + \frac{2N}{R_o^2 C}\right)\left(s + \frac{1}{R_o}\right)} \quad (3)$$

$$C(s) = \frac{Lred}{\left(\frac{s}{k} + 1\right)} \quad (4)$$

where,

δq = Error in Queue length
δp = Error in Probability
C = out-rate of the router in packets/second
Ro = Queuing + One way Propagation delay
Queuing delay = $\frac{q_o}{C}$
Propagation delay = T_p (fixed)
Lred = P_{max} / (max_{th} - min_{th})
P_{max} = maximum probability of dropping a packet.
max_{th} = maximum threshold in packets
min_{th} = minimum threshold in packets
N = no. of flows
alpha = queue averaging parameter

$$k = -C * \log_e(1 - alpha) \quad (5)$$

Thus the overall open loop transfer function is given by

$$G(s) = \frac{Lred * \frac{(R_o C)^3 e^{-sR_o}}{(2N)^2}}{\left(\frac{s}{k} + 1\right)\left(\frac{R_0^2 Cs}{2N} + 1\right)(R_0 s + 1)} \quad (6)$$

Assuming that the system operates in the frequency range less than $\varpi_{gopt} = 0.1 \min\left(\frac{2N}{R_o^2 C}, \frac{1}{R_o}\right)$ (7)

The simplified transfer function becomes $G(s) = \dfrac{Lred * \frac{(R_o C)^3 e^{-sR_o}}{(2N)^2}}{\left(\frac{s}{k} + 1\right)}$ (8)

The Delay Margin of the system is the ratio of the Phase margin to the Crossover frequency of the undelayed open loop transfer function [15]. This is the amount of delay increase the system can tolerate before it goes to instability. The Delay Margin for a stable system is positive and it is negative for an unstable system. For the above system, the Delay Margin, Dm of the system is given by

$$\text{Dm} = \left(\frac{\pi - Tan^{-1}\left(\frac{\varpi_g}{k}\right)}{\varpi_g}\right) - R_o \,, \tag{9}$$

where, $\varpi_g = k * \sqrt{\left(Lred * \frac{(R_o C)^3}{(2N)^2}\right)^2 - 1}$ and other parameters are as defined above.

The steady state Tracking Error is the amount of error between the instantaneous queue and the steady state queue. We ideally want the system to have a lower error for better tracking.

For the above system the steady state Tracking error e_{ss} is defined by [15]

$$e_{ss} = \lim_{s \to 0} s * E(s) \tag{10}$$

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

$$E(s) = \left(\frac{1}{1 + G(s)}\right) * \frac{1}{s}$$

The steady state error is then given by

$$e_{ss} = \frac{1}{1 + G(0)} \tag{11}$$

where, $G(0) = Lred * \frac{(R_o C)^3}{(2N)^2}$

From the above equations we see that as $G(0)$ increases, the Delay Margin of the system decreases. However the Tracking Error decreases with increase in $G(0)$. Thus we have better error minimization performance at the expense of lower stability margins and there is a trade-off between better stability and smaller Tracking Error. There is thus an optimal region where we can operate the system to extract best performance.

3. EFFECT OF TUNING AND GUIDELINES FOR PARAMETER SETTING

We now proceed to analyze the performance of a GEO network with the following parameters. One-way latency, T_p = 250ms, max_{th} = 60 packets, min_{th} = 20 packets, C = 250 packets/second, P_{max} = 0.1, N = 5, alpha = 0.002. The network configuration is show in Figure 19.

For the above case, we plot the Tracking Error and the Delay Margin as a function of the Propagation time T_p in 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 in throughput. The NS result for this case is shown in Figure 5. The NS simulation configuration details are given in Section IV. From Figure 5, we can see the high oscillations in the queue. Since the queue goes to zero often, there is less throughput in the system. We now try to improve the performance of the above system. We want the system to have a Delay Margin that is positive. We thus need to decrease the gain, *G(o)* of the system from equation 9. We do this by increasing N to 30. The Delay Margin in this case is infinite as shown in Figure 4 and the system is expected to oscillate less. This reduction in oscillations means an increase in the throughput in the low-delay regions. The NS-results for this case is shown in Figure 6. We can see the reduction in oscillations, which will result in throughput improvement

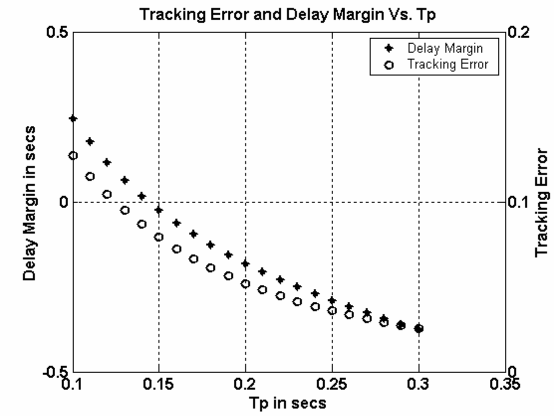

Figure 3. Tracking Error and Delay Margin for unstable Geo Satellite

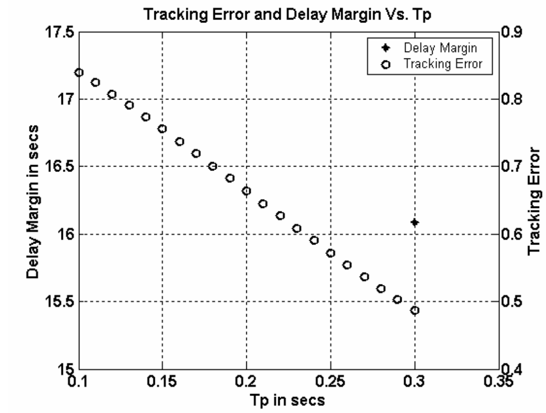

Figure 4. Tracking Error and Delay Margin for stable Geo Satellite

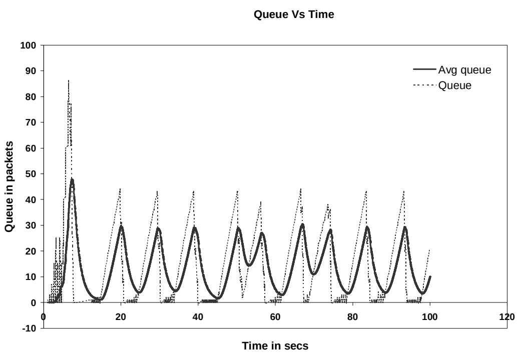

Figure 5. Queue for an unstable Geo Satellite.

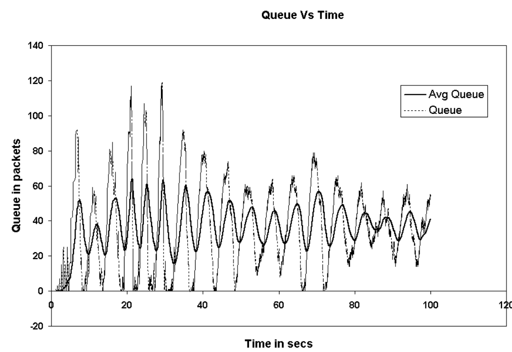

Figure 6. Queue for a stable Geo Satellite, 250 ms

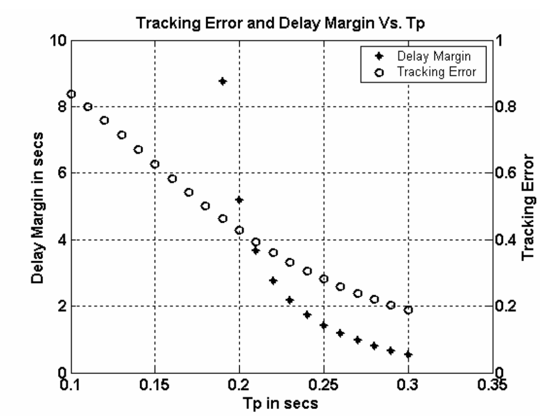

Figure 7. Tracking Error and Delay Margin for lower G(o)

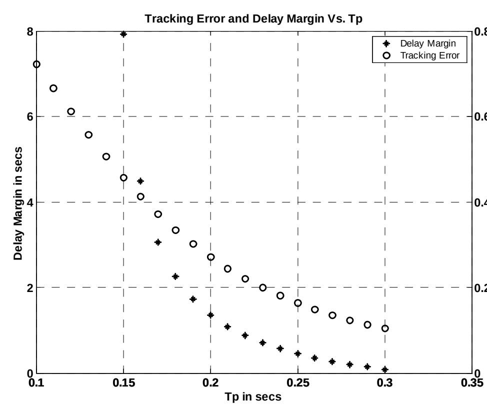

Figure 8. Tracking Error and Delay Margin for higher G(o)

For the given load level N, we obtain the maximum allowable P_{max} that guarantees a positive Delay Margin for system with parameters max_{th} =4, min_{th} = 1, N=30, alpha = 0.002, C = 250 packets/second The maximum value of P_{max} calculated from equation 9 that gives a positive Delay Margin is 0.5. Thus the system is stable for any P_{max} less than 0.5. Having this result we now proceed to tune the system for better performance. Our main goals are stability with minimum tracking error. We test the system in the stable region by changing *G(o)*. A high *G(o)* means a system with reduced error form equation 11. Such a system will give better much performance.

Figures 7 and 8 show the Tracking Error and the Delay Margin as a function of the Propagation time T_p for P_{max} = 0.1 and 0.2. We have decreased Tracking Error when P_{max} is 0.2 compared to 0.1 and are still in the stable region. This means that the throughput of the system is higher with this setting. This is shown by means of NS-simulations in Figure 10, where Throughput Vs. Average Delay for both cases. We can clearly see a performance improvement with proper setting of parameters from equations 9 and 11 in the region where the system is stable.

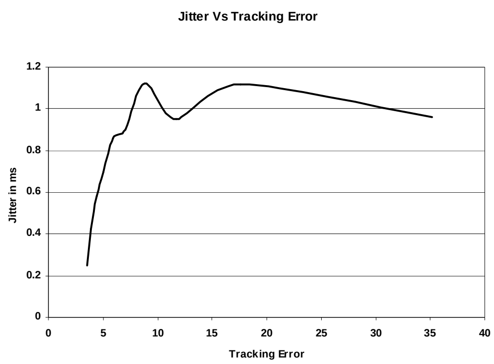

Figure 9. Jitter Vs Tracking Error for GEO satellites

Figure 10. Throughput Vs Avg Delay for 2 different G(o)

We also plot Jitter Vs Tracking Error for the case where P_{max} is 0.1 and the system is stable. We increase the maxth – minth for different cases. An increase in the difference between the thresholds will only increase the Delay Margin for equation 9. But the Tracking error increases from equation 11. We expect the jitter to increase. This is shown in Figure 9. We see that the jitter has indeed gone up with decrease in *G(o)*. This is because by decreasing G(o) we increase the Tracking error and hence the jitter. The jitter is calculated from ns simulations. Thus we can optimize throughput and jitter performance with a proper choice of parameters from equations 9 and 11.

We want to shift the Delay Margin upwards for increased stability, which decreases the Tracking Error and hence lowers performance. We know that $R_o = \frac{q_o}{C} + T_p$. T_p , C being fixed, the Delay Margin is a bound on the queue. Thus if we know an estimate on the variation in the queue, we can tune the parameters such that we can operate with a minimum safe Delay Margin from equation 9 and still be in the low Tracking Error region from equation 11. This will improve performance compared to arbitrarily setting the parameters and hence compromising performance.

For MEO networks, one-way latency = T_p = 120ms, max_{th} = 60 packets, min_{th} = 20 packets, C = 250 packets/second, P_{max} = 0.2, N = 5, alpha = 0.002. For this case, we plot the Tracking Error and the Delay Margin as a function of the Propagation time T_p in Figure 11.

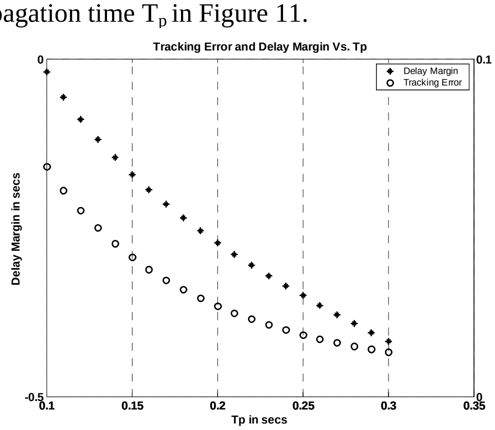

Figure 11. Tracking Error and Delay Margin for unstable MEO network

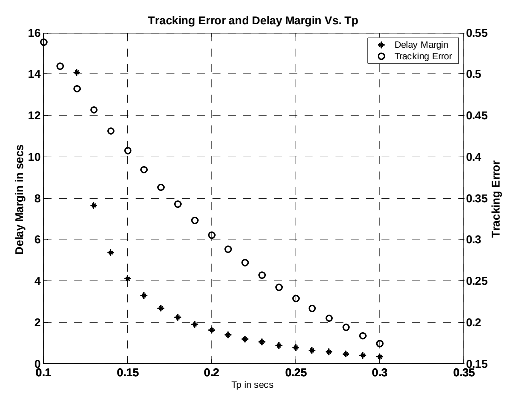

Figure 12. Tracking Error and Delay Margin for stable MEO network

It can be seen that the system has a negative Delay Margin, which means that the system is unstable. The NS results for this case is shown in Figure 13. We can see the high oscillations in the queue. We now try to improve the performance of the above system. We want the system to have a Delay Margin that is positive. We thus need to decrease the gain, *G(o)* of the system. We do this by increasing N to 20. Figure 12 shows that the Delay Margin in this case is infinite and the system is stable. The NS-results for this case is shown in Figure 14. We can see the reduction in oscillations in the queue compared to the first case.

We now show the effect of tuning the parameters in MEO networks. The parameters are One-Way latency (T_p)= 120ms, maxth = 4 packets, min_{th} = 1 packet, C = 250 packets/second, P_{max} = 0.1, N = 10, alpha = 0.002.

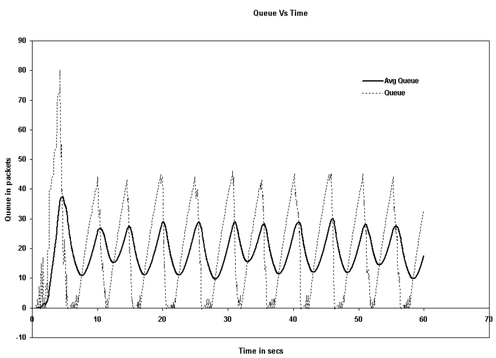

Figure 13.Queue for unstable MEO network

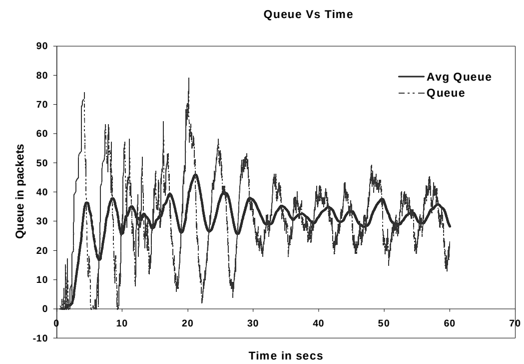

Figure 14. Queue for stable MEO network.

The Tracking Error and Delay Margin plots are shown in Figure 15. The system is stable as the Delay Margin is positive. We now increase the gain in the system by increasing P_{max} to 0.2. The Tracking Error and Delay Margin plots for this case are shown in Figure 16. We see that the Tracking Error has indeed gone down. But the system is still stable Thus we expect the system to give better performance as shown by the NS-plots in Figure 18.

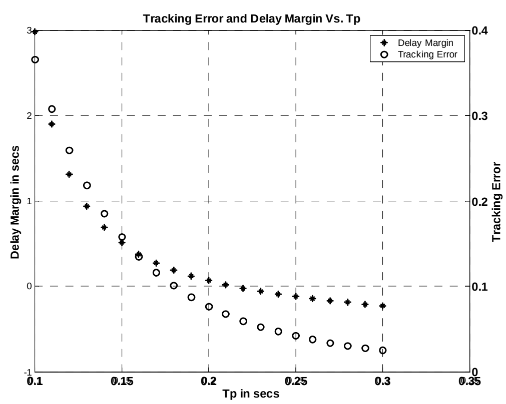

Figure 15. Tracking Error and Delay Margin for lower G(0)

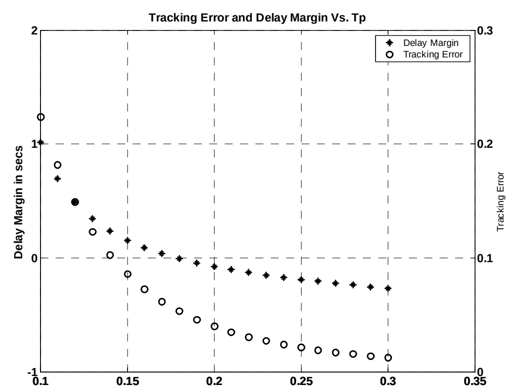

Figure 16. Tracking Error and Delay Margin for higher G(0)

We also plot Jitter Vs Tracking Error for the case where P_{max} is 0.1 and the system is stable. We increase the maxth – minth for different cases. From the jitter argument before in the case of GEO satellites, we see that the jitter has indeed gone up with decrease in *G(o)* as shown in Figure 17. This is because by decreasing *G(o)* we increase the Tracking error and hence the jitter. The jitter is calculated from ns simulations. Thus we can optimize throughput and jitter performance with a proper choice of parameters from equations 9 and 11.

Figure 17. Jitter Vs Tracking Error for LEO satellites

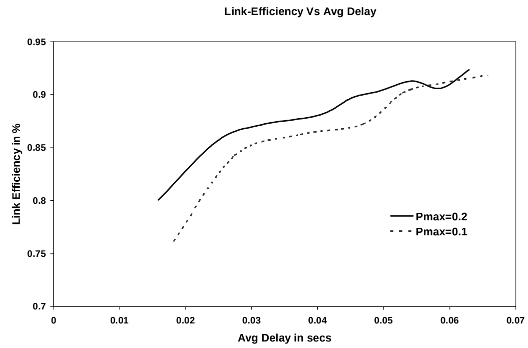

Figure 18. Throughput Vs Avg Delay for 2 different G(0)

4. NS SIMULATIONS - SIMULATION CONFIGURATION

For all our simulations we used the following configuration, shown in Figure 19. 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 2Mbps, Tp ms delay link. R2 is connected to R3 through a 1.5Mbps, Tp ms delay link and a number of destinations D1, D2, D3.., Dn are connected to the router R3 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.

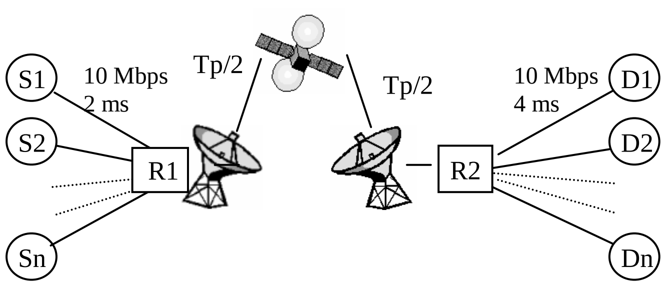

Figure 19. Simulation Configuration

This configuration can simulate the case MEO and GEO satellite networks by varying the delay Tp. A delay of 120 ms and 250ms are used respectively for MEO and 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 is $\alpha = 0.002$. Thus the results obtained by control theory are validated using ns simulations. So our analysis scheme based on control theory can be used to optimized IP networks.

5. CONCLUSIONS

In this paper, we used a previously derived linear model of a TCP-RED scheme to derive margins for the allowable delay in the feedback path of the system. We also performed Tracking Error minimization technique to analyze the performance improvement. We showed that performance improvement occurs with a corresponding decrease in the Delay Margin of the system. We have given design conditions on how to increase performance without violating feedback system stability. Thus using our results a well optimized satellite router could be designed. We have verified our results using NS simulator.

ACKNOWLEDGEMENTS

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

REFERENCES

[1] Arjan Durresi, Sastri Kota, Raj Jain, Mukul Goyal, "Achieving QoS for TCP Traffic in Satellite Networks with Differentiated Services", Proceedings of Journal of Space Communications, Volume 17, Number 1-3, p. 125-136, 2001.

[2] Sastri Kota, "Multimedia Satellite Networks: Issues and Challenges," *Proc. SPIE International Symposium on Voice, Video, and Data Communications, Bosto*n, Nov 1-5, 1998.

[3] A. Durresi, M. Sridharan, C. Liu, M. Goyal, R. Jain, " Congestion Control using Multilevel Explicit Congestion Notification in Satellite Networks," Proceedings of 10th IEEE International Conference on Computer Communications and Networks (ICCCN2001, pp. 483-488, October 14-19, Phoenix, Arizona, 2001.

[4] K. Ramakrishnan and S. Floyd, ``A Proposal to add Explicit Congestion Notification (ECN) to IP," *RFC 248*1, January 1999.

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

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

[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-387-9*9, 1999.

[8] Floyd, S., and Fall, K., Promoting the Use of End-to-End Congestion Control in the Internet, IEEE/ACM Transactions on Networking, August 1999.

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

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

[11] D. Chiu and R. Jain, ``Analysis of the Increase/Decrease Algorithms for Congestion Avoidance in Computer Networks," *Journal of Computer Networks and ISDN System*s, Vol. 17, No. 1, June 1989, pp. 1-14.

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

[13] Chunlei Liu and Raj Jain, "Improving Explicit Congestion Notification with the Mark-Front Strateg*y," Computer Network*s, 35, 2000, pp. 185-201.

[14] NS Simulator, Available from http://www.isi.edu/nsnam/ns.

[15] H. Ozbay, Introduction to Feedback Control Theory, CRC Press LCC, Boca Raton FL, 1999.

[16] 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 Networkin*g, November 1997.

[17] A Durresi, M Sridharan, C. Liu, R. Jain, "Improving Congestion Control in Satellite Networks," Proceedings of SPIE Conference on Quality of Service over the Net Generation Data Networks, Denver, August 20-24, 2001. V. 4524, pp. 293-303.

[18] C. Hollot, V. Misra, D. Towsley, and W. Gong, "A Control Theoretic Analysis of RED." pages 1510-1519, INFOCOM, 2001.

[19] 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.

[20] M. May, J. Bolot, C. Diot, and B. Lyles. Reasons Not to Deploy RED. *Proc. of 7th. International Work-shop on* Quality of Service (IWQoS'99), pages 260–262, June 1999.

[21] T. Ott, T. Lakshman, and L. Wong. SRED: Stabilized RED. INFOCOM, 1999.