Performance Evaluation of Exponential TCP/IP Congestion Control Algorithm

M.Kalpana[1] **and Dr.T.Purusothaman** [2]

[1] Ph.D.Research Scholar, Anna University-Chennai, Government College of Technology, Coimbatore 641013, India.

[2]Assistant Professor, Department of CSE, Government College of Technology, Coimbatore 641013, India

Abstract

This paper introduces and analyses a class of non-linear congestion control algorithms called exponential congestion control algorithm. It generalizes the AIMD algorithms used for the TCP connections. This algorithm provides additive increase using the inverse of an exponential of the current window size and provides multiplicative decrease using the exponential of the current window size. It is further parameterized by α and β. There are infinite numbers of TCP-compatible exponential algorithms by assuming exponent of different order. This paper analyses the performance of the generalized algorithm (TCP/EXP), for the TCP networks. TCP compatibility of this algorithm is evaluated using the simulations of the implementation of the proposed model. Simulations are done using ns2, a discrete event simulator. The results of simulation are compared with that of the TCP variants such as TCP/Tahoe, TCP/Reno, TCP/New Reno, MIMD and PIPD. The Comparison shows that the proposed algorithm performs better in terms of throughput.

Keywords
Congestion control, TCP, Non-linear algorithms, AIMD, ns2, MIMD, PIPD, Exponential Congestion Control Algorithm.

1. Introduction.

Computer networks, both wired and wireless networks, have been growing tremendously in the last decade. Many wired networks, wireless networks, mobile ad hoc networks and wireless sensor networks are being developed which interconnects computers used by both private and public sectors. In most of these networks the protocol stack used for interconnection is TCP/IP. In spite of the rapid growth and explosive increase in traffic demand, computer networks in general, Internet in particular is still working without collapse.

The growth of the Internet has sparked the demand of several applications, which require the stability of the Internet. For achieving such a success and to have the stability of Internet, mechanisms are developed to reduce transmission errors, to provide better bandwidth sharing of sources that use common bottleneck links, to reduce the round trip time (RTT) and mainly to provide the congestion control by the Transport layer protocol i.e. TCP (Transmission Control Protocol). TCP's end-to-end congestion control mechanism reacts to packet loss by adjusting the number of outstanding unacknowledged data segments allowed in the network [1] and [2]. Such algorithms are implemented in its protocol, TCP. [3], [4] and [5]. In the existing algorithms, increasing the congestion window linearly with time increases the bandwidth of the TCP connection and when the congestion is detected, the window size is multiplicatively reduced by a factor of two [11].

TCP is not well suited for several emerging applications including streaming and real time audio and video because it increases end-to-end delay and delay variations [9]. New congestion control algorithms are being developed even now, because of more and more computers get interconnected only using TCP. Two such protocols developed and provides better throughput for the wired and wireless networks are MIMD and PIPD [26], [27], [28] and [29].

This paper analyzes a new class of nonlinear congestion control algorithm for Internet Transport Protocols and applications. It seeks to develop an algorithm for applications such as Internet audio and video that does not react well to rate reductions, because the rate reduction technique used for these applications will result into the degradation in user-perceived quality [5], [6]. This analysis results into get good understanding of TCP-compatible congestion control algorithms by generalizing the Additive Increase and Multiplicative Decrease (AIMD) algorithms. The proposed algorithm is analysed in a simulated wired TCP network.

The proposed algorithm is named as TCP/EXP. It is compared with the TCP variants such as TCP/Tahoe (called as TCP), TCP/Reno, TCP/NewReno, MIMD and PIPD and the comparisons are simulated in ns-2, the discrete event driven simulator that is used by most of the network researchers.

This paper is structured as follows. Section 2 briefly describes the basic AIMD congestion control rules and the existing TCP congestion control algorithms. Section 3 discusses about the proposed generalized

---

Manuscript received March 5, 2009
Manuscript revised March 20, 2009

congestion control algorithms called Exponential algorithm – TCP/EXP. Section 4 analyses the window size variation with respect to time of the proposed algorithm. Section 5 simulates a wired TCP network with Dumbbell topology and with different sources employing the various congestion control mechanisms. The paper finally concludes in section 6.

2. Window Based Congestion Control for TCP Networks.

The state of art of the network congestion shows that it is a very difficult problem because there is no way to determine the network condition. The congestion occurs when there is a lot of traffic in the networks. Rapidly increasing bandwidths and great variety of software applications have created a recognized need for increased attention to TCP congestion control mechanisms [22].

TCP is a connection-oriented protocol that offers reliable data transfer as well as flow and congestion control. [5]. TCP maintains a congestion window that controls the number of outstanding unacknowledged data packets in the network. Sending data consumes slots in the window of the sender and the sender can send packets only as long as free slots are available [21].

On start-up, TCP performs slowstart, during which the rate roughly doubles each round-trip time to quickly gain its fair share of bandwidth [12]. In steady state, TCP uses the AIMD mechanism to detect additional bandwidth and to react to congestion. When there is no indication of loss, TCP increases the congestion window by one slot per round-trip time. In case of packet loss, indicated by a timeout, the congestion window is reduced to one slot and TCP reenters the slowstart phase. Packet loss indicated by three duplicate ACKs results in a window reduction to half of its previous size.

The AIMD algorithm may be expressed as given in [9], [10] and [11].

$$\begin{aligned} &I:\ W_{t+R} \leftarrow W_t + \alpha\ ;\ \alpha > 0 \\ &D:\ W_{t+\delta t} \leftarrow (1-\beta)\ W_t; \\ &0<\beta<1 \end{aligned} \qquad (1)$$

where

- $I \rightarrow$ Increase in window as a result of the receipt of one window of acknowledgement in a round-trip-time (RTT)
- $D \rightarrow$ Decrease in window size on detection of congestion by the sender
- $W_t \rightarrow$ Window size at time t
- $R \rightarrow$ RTT of the flow and
- α and $\beta \rightarrow$ Increase and Decrease Rule constants.

There are many variations to these algorithms so as to gain more bandwidth by adjusting the window size[22],[25]. The ns-2 simulator have implementations of many such variants [15]. The algorithm represented by equation (1) is implemented as TCP/Tahoe , simply called as TCP [11]. The other variants implemented are TCP/Reno, TCP/NewReno, TCP/Fast, TCP/SACK etc. These algorithms vary in slowstart and congestion avoidance phases [28].

3. Exponential TCP Congestion Control Algorithms

This section discusses the properties of the proposed exponential congestion control algorithms. The window adjustment policy is only one component of the congestion control protocol derived from exponential algorithm. Other mechanisms such as congestion detection (loss, ECN etc.), retransmissions (if required), estimation of Round-trip-time etc., remain the same as TCP [8]. The proposed algorithms mainly aims in increasing the window size faster and to gain the bandwidth quicker.

The authors generalized the AIMD rules in the following manner, in order to study and understand the notions of TCP – compatibility and the trade off between the increase and decrease rules.

The exponential rules are given below:

$$\begin{aligned} &I:\ W_{t+R} \leftarrow W_t + (\alpha / e^{W_t})^{k} \\ &D:\ W_{t+\delta t} \leftarrow W_t - (\beta\ e^{W_t})^{l} \end{aligned} \qquad (2)$$

By choosing different values of α and β in equations (2), they became the members of the exponential family. By including the higher order terms, exponential increase and decrease algorithms of different orders are obtained that may be used for the window size adjustment for the congestion avoidance phase.

4. Analysis of the Algorithm : TCP/Exp

The exponential algorithm is implemented and represented by equation (2). It is represented as TCP/EXP in section 3. This algorithm is implemented to study the variation of window size and the resulting throughput with respect to time. The algorithm begins in the slowstart state [12]. In this state, the congestion window size is doubled for every window of packets acknowledged. Upon the first congestion indication, the congestion window size is cut in half and the session enters into the exponential congestion control state [11],[21] and [22].

In this state the congestion window size is increased by $(\alpha /e^{Wt})^{k}$ for each new acknowledgement received, where W_t is the current congestion window size. The algorithm reduces the window size when congestion is detected. Congestion is detected by two events: (i) triple-duplicate ACK and (ii) time-out. If by triple-duplicate ACK, the algorithm reduces the window size by $(\beta e^{Wt})^{l}$. If the congestion indication is by time-out, the window size is initialized to 1.

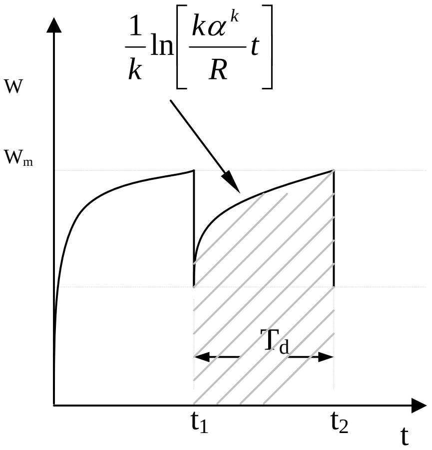

Fig 1. Functional form of window vs. time curve

The window size and throughput of this algorithm is compared with the standard congestion algorithms for TCP. [9], [21] and [25]. The authors have used the TCP variants available in ns-2 such as TCP/Tahoe, TCP/Reno and TCP/NewReno for comparison [15]. The Polynomial algorithms MIMD and PIPD proposed in [26], [27], [28] and [29] is also used for comparison.

Using the Increase rule of TCP/EXP algorithm and by linear interpolation of window size between Wt and W_{t+R} is given by,

$$\frac{dW}{dt}=\frac{\alpha^{k}}{e^{W_t k}R} \tag{3}$$

$$\frac{e^{W_t k}}{k}=\frac{\alpha^{k}t}{R}+C \tag{4}$$

where C is the constant of integration. Leaving the constant of integration i.e. assuming C=0, the equation may be written as

$$e^{W_t k}=\ln\left[\frac{k\alpha^{k}}{R}t\right] \tag{5}$$

$$W_t=\frac{1}{k}\ln\left[\frac{k\alpha^{k}}{R}t\right] \tag{6}$$

The functional form of the curve given by equation 6 is shown in figure 1.

5. Performance Evaluation of Exponential Algorithms in Wired TCP Networks

This section presents the results of the ns-2 simulation of exponential algorithm in wired TCP networks [15]. The authors start by investigating the connections running the TCP-compatible exponential algorithm TCP/EXP. The simulations use the topology shown in figure 2.

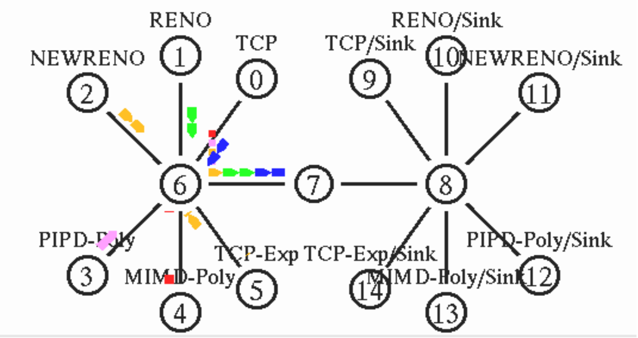

Fig .2 Dumbbell Topology used for wired TCP network

It consists of 6 connections sharing a bottleneck link with total bandwidth equal to b, where all connections have an almost identical round-trip propagation delay equal to RTT. Each exponential flow uses a modified TCP with AIMD algorithm replaced by the exponential window size variations given by equation (2); other mechanisms like slow-start and time-out remain unchanged as already mentioned in section 4 of this paper. Each source always has data to send, modeled using the network simulators “FTP” application.

The following simulations are carried out:

- Simulation 1: Varying buffer size and varying simulation time in two buffer management schemes – DropTail and RED
- Simulation 2: Performance comparison of congestion control algorithms with ten different buffer management schemes.

5.1 Simulation 1

Two buffer management schemes – DropTail and RED are assumed at the routers at the nodes 6, 7 and 8. With these buffer management schemes two different simulating conditions are assumed.

- Simualtion time is fixed at 100 seconds. The buffer size (in number of data Packets) is varied as 25, 30, 35, 40, 45 and 50.
- Buffer size is fixed at 50. The simulation time is varied as 50, 100, 150, 200, 250 and 300 seconds.

In both these cases, the number of data packets received by the sinks corresponding to the TCP sources with the implementation of different congestion control algorithms such as TCP/Tahoe, TCP/Reno, TCP/NewReno, MIMD, PIPD and TCP/EXP are recorded. This information is available in the trace file generated during the simulation. Using this trace file, the throughput corresponding to each source node is also calculated using the equation

$$\lambda = \frac{c \cdot s}{R \cdot \sqrt{p}} \qquad (7)$$

where s is the segment size, R *is* the round trip time, p is the packet loss rate and c is a constant value commonly approximated as $1.5\sqrt{2/3}$. The values of s, R and p are available in the trace file.

The values of data packets received and throughput calculated are plotted for the two simulating conditions mentioned above.

Figure 3 depicts the data packets (in bytes) received by the TCP/Sink nodes corresponding to the TCP/Source nodes implemented with the different congestion control algorithms. The buffer size is varied assuming the Simulation time as constant (100sec.), with the DropTail buffer management scheme. Figure 4 gives the throughput of these six TCP/Sources with the same simulating environment. Both the figures show that the TCP/EXP algorithm performs better compared to the other TCP Congestion control algorithms.

Figures 7 and 8 shows the data received and throughput calculated using equation 7 assuming the RED buffer management schemes. The other simulation environment is the same as described above.

Figure 5 plots the values of data packets received by the TCP/Sinks from the different sources assuming the buffer size as constant (50). The Simulation time in these are varied as 50, 100, 150, 200, 250 and 300 seconds. Figure 6 plots the throughput calculated using the equation 7 and the trace file generated for the same simulation.

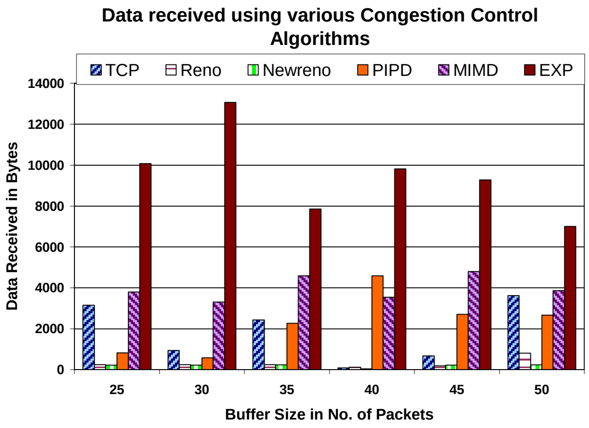

Figure 3. Data received with DropTail Buffer Management with varying Buffer Size

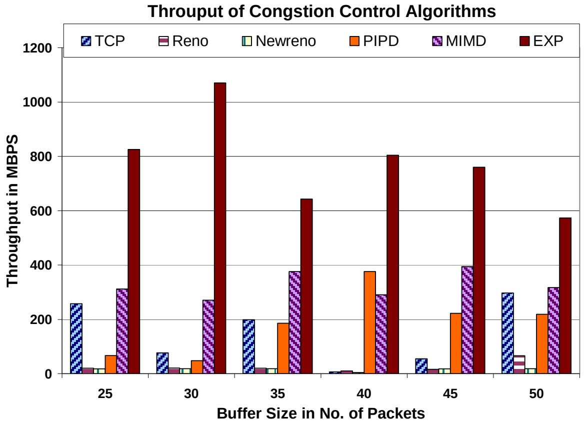

Figure 4. Throughput with DropTail Buffer Management with varying Buffer Size

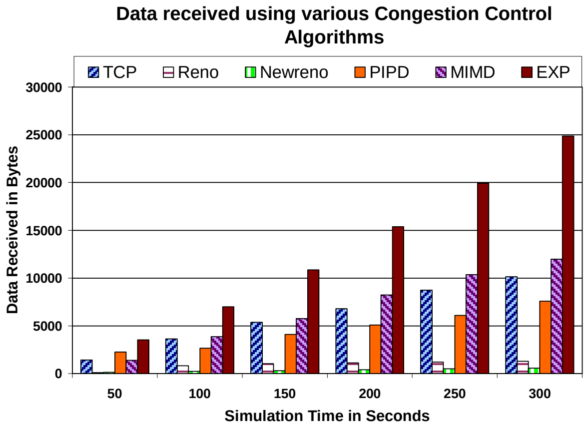

Figure 5. Data received with DropTail Buffer Management with varying Simulation Time

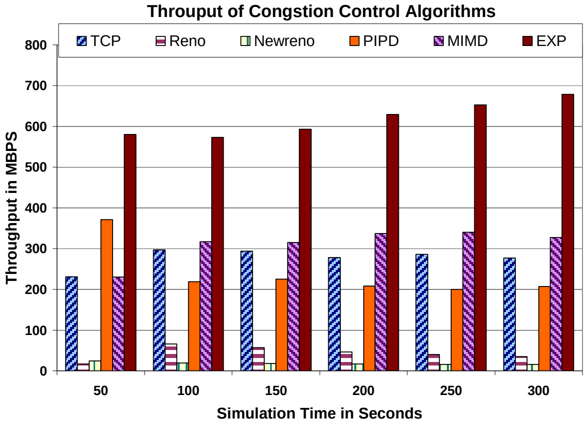

Figure 6. Throughput with DropTail Buffer Management with varying Simulation Time

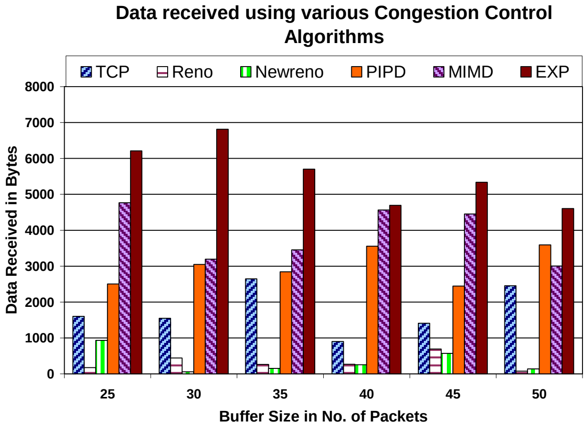

Figure 7. Data received with RED Buffer Management with varying Buffer Size

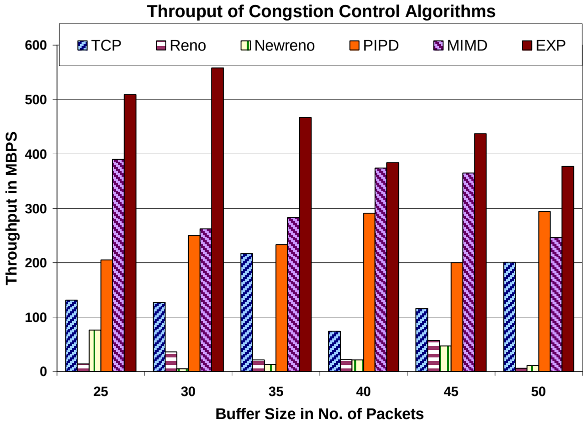

Figure 8. Throughput with RED Buffer Management with varying Buffer Size

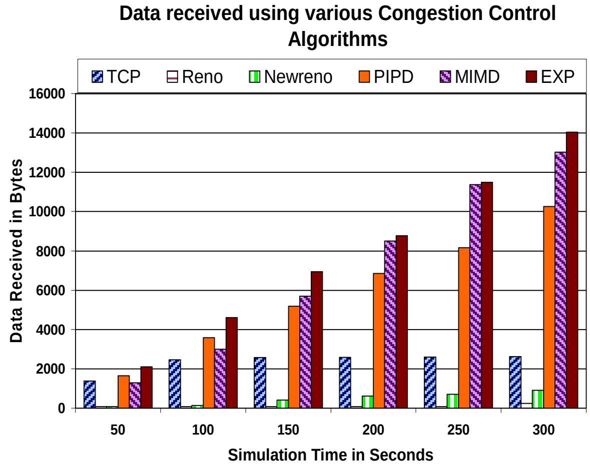

Figure 9. Data received with RED Buffer Management with varying Simulation Time

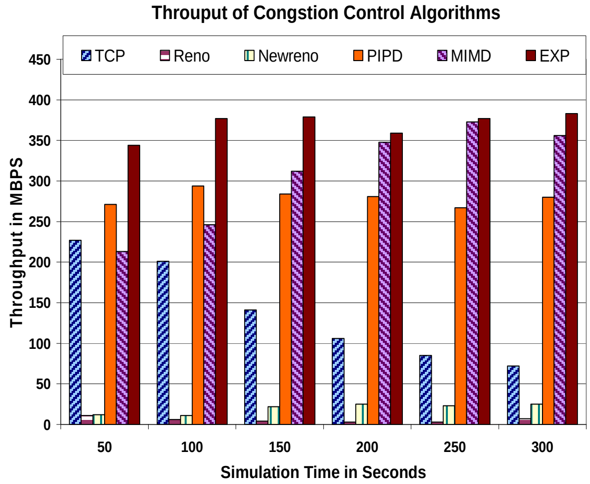

Figure 10. Throughput with RED Buffer Management with varying Simulation Time

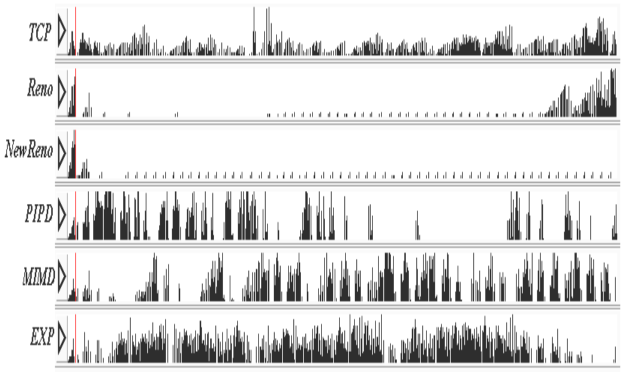

Figure 11. Bandwidth versus time graph for the six congestion control algorithms

Figures 9 and 10 shows the data received and throughput, assuming the RED buffer management scheme in the above simulation with all other parameters assumed to be the same. The results plotted from all these simulations show that the proposed Congestion control algorithm TCP/EXP outperforms other TCP congestion control algorithms in terms of throughput and number of data packets received.

Figure 11 shows link bandwidth due to each congestion control algorithm during the entire period of simulation in the case of DropTail buffer management technique with buffer size of 50 data packets and a simulation time of 100 seconds. This figure shows that the TCP/Exp congestion control algorithm utilizes the link more efficiently and hence the link bandwidth is more compared with other congestion control algorithms.

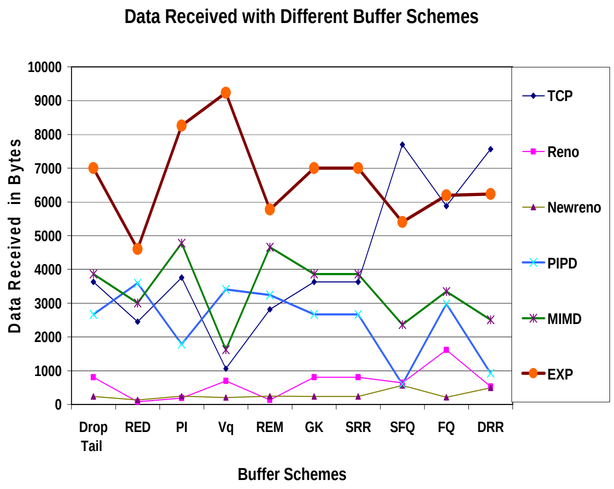

Figure 12. Data Received in Different Buffer Management Schemes

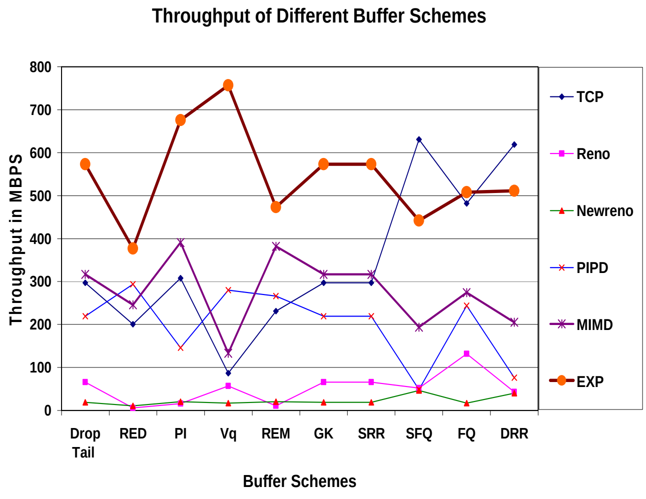

Figure 13. Throughput in Different Buffer Management Schemes

5.2 Simulation 2

With the topology simulated as shown in figure 2, the router is assumed to be implemented with 10 different buffer management schemes. The schemes for implementation at the routers in the nodes 6, 7 and 8 are explained briefly below.

1. Drop Tail: This scheme implements FIFO scheduling and drop-on-overflow buffer management which is typical of most present-day Internet routers.

2. RED (Random Early Drop): This scheme regulates the queue length to a desired value by adapting the marking probability. This marks each packet with a probability p, which is periodically updated.

3. FQ (Fair Queuing): Implements sharing of the buffer equally among all the contending sources and places the packets fairly in the per-flow queue.

4. SFQ (Stochastic Fair Queuing): Router uses a hashing function to put a flow to its correct position in the per-flow queue.

5. PI (Proportional Integral Controller): This scheme marks each packet with a probability p and the value of p is periodically updated. This is similar to the RED scheme. But the update equations are different

6. Vq (Virtual Queue): This scheme maintains a virtual queue whose capacity is less than the actual capacity of the link. When a packet arrives in the real queue, the virtual queue is also updated to reflect the new arrival. Packets in the real queue are marked/dropped when virtual buffer overflows.

7. REM (Random Exponential Marking): The marking probability depends on the sum of link prices (Congestion Control), summed over all the routers in the path.

8. GK (Gribbens-Kelly Virtual Queue): The link maintains a virtual queue with a buffer size of B'=kB and k<1, where B is the buffer size of original queue. When the virtual queue overflows, all the packets in the real queue and all future incoming packets are marked till the virtual queue becomes empty again.

9. DRR (Deficit Round Robin): The flows arrive at the router are queued in input buffer and wait for an en-queue action. The en-queue finds the right queue for each flow according to its IP source and destination address pair. Then the packets of flows in the correct queue are released to the output buffer according to the round robin rule.

10. SRR (Stochastic Round Robin): This is similar to the weighted round robin but adjusted to account for variable sized packets, when surplus (unused bandwidth) is carried on to the next round.

The Simulation is assumed with the buffer size of 50 data packets and simulation time of 100 seconds. Figure 12 is plotted with the number of data packets received with each buffer management scheme assuming the six congestion algorithms TCP/Tahoe, TCP/Reno, TCP/NewReno, MIMD, PIPD and TCP/EXP implemented in the TCP source nodes. Figure 13 shows the throughput for the same ten buffer management schemes and six congestion control algorithms. The figures show that the proposed congestion control algorithm TCP/EXP performs better with all the management schemes except SFQ and DRR. With these two buffer management schemes, TCP/Tahoe performs better. Next to TCP/EXP, MIMD performs better in almost all the buffer management techniques assumed for simulation.

6. Conclusion

This paper presents and evaluates a family of nonlinear congestion control algorithms, called exponential algorithm such as TCP/EXP. This exponential algorithm generalizes the familiar class of linear algorithms.

The simulation results showed good performance and interactions between exponential algorithm (TCP/EXP) and TCP using standard algorithms in wired networks. The algorithm TCP/EXP obtains higher throughput than the standard algorithms for TCP, TCP/Reno, TCP/NewReno, MIMD and PIPD. Figures 3 to 10 show the results of simulation assuming the two buffer management techniques – DropTail and RED, with varying buffer size and varying simulation time.

All these results show that TCP/EXP performs better than the other congestion control algorithms. Figures 12 and 13 show that the TCP/EXP performs better compared with the other algorithms in most of the buffer management schemes assumed at the routers. This paper clearly indicates that variation in the basic AIMD congestion control algorithm such as the proposed TCP/EXP algorithm, will definitely improve the transmission efficiency in wired TCP networks.

References

[1] Van Jacobson, "Congestion Avoidance and Control," *ACM Computer Communication Review*, vol.18, pp. 314--329, Aug. 1988. *Proceedings of the Sigcomm '88 Symposium* in Stanford, CA, August 1988.

[2] J Widmer, R Denda, and M Mauve. “A Survey on TCP-Friendly Congestion Control.” *IEEE Network Magazine*, May 2001.

[3] Congestion Control Schemes for TCP/IP Networks current implementation in BSD Unix. Douglas E. Comer, *Internetworking with TCP/IP Volume I*, Englewood Cliffs, New Jersey, Prentice Hall, 1991.

[4] W. R. Stevens, *TCP/IP Illustrated, Volume 1*, Addison-Wesley, Reading, MA, November 1994.

[5] M. Allman and V. Paxson, *TCP Congestion Control*, Internet Engineering Task Force, April 1999, RFC 2581.

[6] Sally Floyd and Kevin Fall, “Promoting the use of end-to-end congestion control in the Internet,” *IEEE/ACM Transactions on Networking*, vol. 7(4), pp. 458–472, August 1999.

[7] R. Rejaie, M. Handley, and D. Estrin, “RAP: An End-to-end Rate-based Congestion Control Mechanism for Real time Streams in the Internet,” in *Proc. IEEE INFOCOM*, March 1999, vol. 3, pp. 1337–1345.

[8] Sally Floyd, "TCP and Explicit Congestion Notification," *ACM Computer Communication Review*, vol. 24, pp. 8-23, Oct. 1995.

[9] Shudong Jin, Liang Guo, Ibrahim Matta, and Azer Bestavros, “A spectrum of TCP-friendly window-based congestion control algorithms,” Tech. Rep. BU-CS-2001-015, Computer Science Department, Boston University, July 2001, Available at http://www.cs.bu.edu/techreports/2001-015-spectrum-tcp-friendly.ps.Z.

[10] Deepak Bansal and Hari Balakrishnan, “Binomial congestion control algorithms,” in *Proceedings of IEEE INFOCOM*, April 2001.

[11] Y. Richard Yang and Simon S. Lam, “General AIMD congestion control,” in *Proceedings of ICNP*, November 2000.

[12] Zheng Wang and Jon Crowcroft, "A New Congestion Control Scheme: Slow Start and Search (Tri-S)," *ACM Computer Communication Review*, vol. 21, pp 32-43, Jan 1991

[13] Lawrence S. Brakmo and Sean W. O'Malley, "TCP Vegas: New Techniques for Congestion Detection and Avoidance," in *SIGCOMM '94 Conference on Communications Architectures and Protocols*, (London, United Kingdom), pp. 24-35, Oct. 1994.

[14] D. Bansal and H. Balakrishnan, “TCP-friendly Congestion Control for Real-time Streaming Applications,” Tech. Rep. MIT-LCS-TR-806, MIT Laboratory for Computer Science, May 2000

[15] “ns-2 Network Simulator,” http://www.isi.edu/nsnam/ns/, 2000.

[16] S. Floyd and V. Jacobson, “Random Early Detection Gateways for Congestion Avoidance,” *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, Aug. 1993

[17] Tejas Kokje Vijay Kakadia, “ Analysis of Congestion Control Strategies For TCP Variants Using Droptail and RED Queuing Disciplines Department of Computer Science University of Southern California, Los Angeles

[18] K.Sundaresan, V.Anantharaman, H-Y Hsieh, and R.Sivakumar. “ATP: a reliable transport protocol for ad hoc networks”, *ACM Mobihoc*, Jun.2003.

[19] G.Holland, and N.H.Vaidya, "Analysis of TCP performance over mobile ad hoc networks", *ACM MOBICOM '99*, Seattle, August 1999.

[20] K.Chen, Y.Xue, and K.nahstedt, "On setting TCP's congestion window limit in mobile ad hoc networks", *IEEE ICC'03*, Anchorage, Alaska, May 2003.

[21] Kevin Fall and Sally Floyd, "Simulation-based Comparisons of Tahoe, Reno, and SACK TCP." *Computer Communications Review*, July 1996

[22] Georgi Kirov," A Simulation Analysis of the TCP Control Algorithms",*International Conference on Computer Systems and Technologies* - ***CompSysTech' 2005 .***

[23] K. Seada, S. Gupta, and A. Helmy. "Systematic Evaluation of Multicast Congestion Control Mechanisms." *SCS SPECTS*, July 2002.

[24] K. Seada and A. Helmy. "Fairness Evaluation Experiments for Multicast Congestion Control Protocols." *Technical Report 02-757,University of Southern California, CS Department,* March 2002

[25] Bogdan Moraru, Flavius Copaciu, Gabriel Lazar and Virgil Dobrota, "Practical Analysis of TCP Implementations: Tahoe, Reno, NewReno", *Technical University of Cluj-Napoca*

[26] Chandrasekaran M. and Wahida Banu R.S.D., 'Performance Evaluation of Polynomial Congestion control Algorithms MIMD-Poly and PIPD-Poly in TCP networks', Asian Journal of Information Technology, Vol.5, No.3, pp.346-357, 2006.

[27] Chandrasekaran M. and Wahida Banu R.S.D., ' Interaction Between Polynomial Congestion Control algorithms Queue Management schemes in wired TCP Networks,' International Journal of Soft Computing, Vol. 1, No. 2, pp.83-90, 2006.

[28] Chandrasekaran M, Kalpana M and Wahida Banu R.S.D., 'Congestion Control using Polynomial Window size adjustment Algorithms for wired and wireless networks', In the Proceedings of International Conference on Network – ICN06 conducted at Mauritius, April 2006. This paper is now available in the Conference Proceedings by IEEE Computer Society and on IEEE Xplore.

[29] Chandrasekaran M, , Kalpana M and Wahida Banu R.S.D., 'Interaction between MIMD-Poly & PIPD-Poly Algorithms and other TCP Variants in Multiple Bottleneck TCP Networks', In the Proceedings of IEEE Conference WOCN2006, Bangalore, April 2006. This paper will be available on IEEE Xplore.

BIOGRAPHICS

M. Kalpana received her B.E. degree in Electronics and Communication Engineering from P.S.G. College of Technology, Coimbatore under Bharathiar University. She did her M.E. in Communication Systems under Anna University, Chennai. She has registered her Ph.D. in Information and Communication Engineering in Anna University, Chennai and presently doing her research work in the field of Wireless Sensor Networks. She will be submitting her Thesis by June 2009. She has 14 years of teaching experience. She has submitted technical papers in 8 National conferences and 8 International conferences including 5 IEEE conferences. She has published 4 papers in International Journals. Her research interests include wireless sensor networks, congestion control in mobile ad hoc networks etc., She is a life member of ISTE. She is a member of IEEE. She has guided many B.E., and M.E., Project works for the students.

Dr.T.Purusothaman, completed his BE degree in the year 1988 under Madras University and completed his ME(CSE) degree at Government College of Technology, Coimbatore in the year 2002. He did his Ph.D. under Anna University in 2006. He has 21 years of teaching experience. He is currently working as Assistant Professor in the Department of Computer science and Engineering at Government College of Technology, Coimbatore. His research areas include Network Security and cryptography, Distributed operating systems, Mobile computing, Advanced Genetic Algorithm and Grid Computing. He has published ten Research articles in reputed Indian and International Journals and fifty Research articles in national and international conferences. He is currently supervising 20 Ph.D., Research Scholars. He is a life member of ISTE. He served as member of Inspection committee representing DOTE, Tamil Nadu for inspecting Polytechnic colleges. He served as a member of Academic council of various Universities. He guided many B.E., M.E. and M.C.A Projects. He is the co-investigator of the project titled "Optimization of search time for evaluation of ciphers using Genetic Algorithm based cryptanalysis" sponsored by Ministry of Communications & Information Technology, New Delhi. He has presented a paper in International conference and attended a Faculty Development Program at Canada.