Asian Journal of Information Technology 5 (3) : 346-357, 2006
© Medwell Online, 2006

Performance Evaluation of Polynomial Congestion Control Algorithms Mimd-Poly and PIPD-Poly in TCP Networks

M. Chandrasekaran and R.S.D.Wahida Banu
Department of ECE, [2]Department of CSE, Government College of Engineering, Salem 636 011, India

Abstract: This study introduces and analyses a class of non-linear congestion control algorithms called polynomial congestion control algorithms. They generalize the AIMD algorithms used for the TCP connections. These algorithms provide additive increase using a polynomial of the inverse of the current window size and provide multiplicative decrease using the polynomial of the current window size. They are further parameterized by α and β. There are infinite numbers of TCP-compatible polynomial algorithms by assuming polynomial of different order. This paper analyses the performance of two models (named as MIMD-Poly and PIPD-Poly) of these generalized algorithms, for the wired (with unicast and multicast) and wireless TCP networks. TCP compatibility of these algorithms is evaluated using the simulations of the implementations of the proposed two models. Simulations are done using ns2, a discrete event simulator. The model MIMD-Poly is proved to be TCP-compatible. The results of simulation are compared with that of the TCP variants such as TCP/Tahoe. TCP/Reno, TCP/New Reno and TCP/Fast. The Comparison shows that both algorithms perform better in terms of throughput.

Key words: Congestion control, TCP, non-linear algorithms, AIMD, ns2, Mobile Ad Hoc networks, multicast

INTRODUCTION

During the last decade, computer networks have been growing very tremendously. Large numbers of computers get connected to both private and public networks. In most of these networks the protocol stack used is TCP/IP. In spite of the rapid growth and explosive increase in traffic demand, computer networks in general, Internet in particular are still working without collapse.

Also 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[1,2]. Such algorithms are implemented in its protocol, TCP[3-5]. 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[6].

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[7,8].

In this study we present and analyze a new class of nonlinear congestion control algorithms for Internet Transport Protocols and applications. We seek to develop a family of algorithms 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]. We also try to get good understanding of TCP-compatible congestion control algorithms by generalizing the Additive Increase and Multiplicative Decrease (AIMD) algorithms. We analyze the proposed algorithms in a simulated wired TCP network.

One of the current challenges of the Internet is to allow universal access to multimedia transmissions, even for receivers located within networks of different bandwidth and other characteristics. Multicast allows one single transmission to be delivered to a large number of receivers over a network[9]. Congestion control is a major requirement for multicast to be deployed in the current Internet. In this study we have analyzed the performance of the proposed congestion control algorithms in a wired network that employ multicast routing strategies.

Corresponding Author: M. Chandrasekaran, Department of ECE, Government College of Engineering, Salem 636 011, India

With the proliferation of mobile computing devices, the demand for continuous network connectivity regardless of physical location has created greater interest in the use of mobile Ad Hoc networks[10,11]. In this study we also analyze the performance of the proposed algorithms in Mobile Ad Hoc Networks that uses TCP. We compare the performance of the two proposed models with the standard AIMD algorithms implemented for TCP networks.

In all the simulations, the proposed two models, which we have named as MIMD-Poly and PIPD-Poly are compared with the TCP variants such as TCP/Tahoe (called as TCP), TCP/Reno, TCP/NewReno and TCP/Fast. For the simulations we have used ns-2, the event driven simulator that is used by most of the network researchers.

WINDOW BASED CONGESTION CONTROL FORTCP 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[12].

TCP is a connection-oriented protocol that offers reliable data transfer as well as flow and congestion control. 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[13].

On start-up, TCP performs slowstart, during which the rate roughly doubles each round-trip time to quickly gain its fair share of band width[14,15]. In steady state, TCP uses the AIMD mechanism to detect additional band width 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[6,8,16].

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

Where

- I → Increase in window as a result of the receipt of one window of acknowledgement in a Round-Trip-Time (RTT) and
- D → Decrease in window size on detection of congestion by the sender
- W_t → Window size at time t
- R → RTT of the flow and
- α and β → Increase and Decrease Rule constants.

There are many variations to these algorithms so as to gain more bandwidth by adjusting the window size[12,17]. The ns-2 simulator has implementations of many such variants[18]. The algorithm represented by Eq. 1 is implemented as TCP/Tahoe, simply called as TCP[6]. The other variants implemented are TCP/Reno, TCP/NewReno, TCP/Fast, TCP/SACK etc. These algorithms vary in slowstart and congestion avoidance phases.

POLYNOMIAL CONGESTION CONTROL ALGORITHMS

In this study we discuss the properties of the proposed polynomial congestion control algorithms. We note that the window adjustment policy is only one component of the congestion control protocol derived from polynomial algorithms. Other mechanisms such as congestion detection (loss, ECN etc.), retransmissions (if required), estimation of Round-trip-time etc., remain the same as TCP[19]. The proposed algorithms mainly aim in increasing the window size faster and to gain the bandwidth quicker.

We generalize 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 polynomial rules are given below:

$$\text{I: } W_{t+R} \leftarrow W_t + (\alpha/W_t)^k + (\alpha/W_t)^{2k} + (\alpha/W_t)^{3k} + \ldots \text{ ; } \alpha > 0$$
$$\text{D: } W_{t+\delta t} \leftarrow W_t - (\beta/W_t)^l - (\beta/W_t)^{2l} - (\beta/W_t)^{3l} - \ldots \text{ ; } 0 < \beta < 1 \quad (2)$$

These rules generalize the class of all congestion control algorithms based on window size adjustment. Now for the analysis of the algorithms we restrict the above rules into a Model as given below.

$$\text{I: } W_{t+R} \leftarrow W_t + (\alpha/W_t)^k$$
$$\text{D: } W_{t+\delta t} \leftarrow W_t - (\beta/W_t)^l \qquad (3)$$

We named these rules as MIMD-Poly. For various values of k and l the above rules show the forms of various increase and decrease rules
For k = 0, l = 1 the rules show AIMD.

$$I: W_{t+R} \leftarrow W_t + 1$$
$$D: W_{t+\delta t} \leftarrow W_t - \beta W_t \qquad (4)$$

For k = -1, l = 1 the rules show Multiplicative Increase and Multiplicative Decrease (MIMD)

$$I: W_{t+R} \leftarrow W_t + W_t / \alpha$$
$$D: W_{t+\delta t} \leftarrow W_t - \beta W_t \qquad (5)$$

For k = -1, l = 0 the rules show Multiplicative Increase and Additive Decrease (MIAD)

$$I: W_{t+R} \leftarrow W_t + W_t / \alpha$$
$$D: W_{t+\delta t} \leftarrow W_t - 1 \qquad (6)$$

For k = 0, l = 0 the rule show Additive Increase andAdditive Decrease (AIAD)

$$I: W_{t+R} \leftarrow W_t + 1 \qquad (7)$$
$$D: W_{t+\delta t} \leftarrow W_t - 1$$

If we include the higher order terms we get thepolynomial algorithms of different order. For example, weassume the general rules restricted to the first two terms only(second order polynomial algorithms) as shown below:

$$I: W_{t+R} \leftarrow W_t + (\alpha/W_t)^k + (\alpha/W_t)^{2k} \ ; \alpha > 0$$
$$D: W_{t+\delta t} \leftarrow W_t - (\beta /W_t)^l - (\beta /W_t)^{2l} \ ; \ 0 < \beta < 1 \qquad (8)$$

We named these rules as PIPD-Poly. Using the abovesecond order polynomial rules and assuming various values ofk and l we get the following general polynomial algorithms:
For k = 0, l = 0 we get

$$I: W_{t+R} \leftarrow W_t + 2$$
$$D: W_{t+\delta t} \leftarrow W_t - 2 \qquad (9)$$

These rules show AIAD.

For k = 1, l = 1 we get

$$I: W_{t+R} \leftarrow W_t + (\alpha/W_t) + (\alpha/W_t)^2$$
$$D: W_{t+\delta t} \leftarrow W_t - (\beta /W_t) - (\beta /W_t)^2 \qquad (10)$$

These rules show PIPD.
For k = 0, l = 1 we get Additive Increase and PolynomialDecrease (AIPD) and
For k = 1, l = 0 we get Polynomial Increase and AdditiveDecrease (PIAD).

By choosing different values of α and β in Eq. 3 and 8, they became the members of the polynomialfamily. If we include the higher order terms, then we can get polynomial increase and decrease algorithms of differentorders. So we could get all possible algorithms that may beused for the window size adjustment for the congestion avoidance.

ANALYSIS OF THE ALGORITHMS: MIMD-Polyand PIPD-Poly

We have implemented the Algorithms represented by Eq. 3 and 8 and we call these polynomial algorithms MIMD-Poly and PIPD-Poly. Both these algorithms are implemented to study the variation of window size and the resulting throughput with respect to time. The algorithm begins in the slow start state[14]. 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 entersinto the polynomial congestion control state[6,12,13].

In this state the congestion window size is increasedby $[(\alpha/W_t)^k + (\alpha/W_t)^{2k} + (\alpha/W_t)^{3k} + ...]$ for each new acknowledgement received, where W_t is the current congestion window size. The algorithm reduces the windowsize when congestion is detected. Congestion is detected bytwo events: (i) triple-duplicate ACK and (ii) time-out. If bytriple-duplicate ACK, the algorithm reduces the window sizeby $[(\beta W_t)^l - (\beta W_t)^{2l} - (\beta W_t)^{3l} - . . . . .]$. If the congestion indication is by time-out, the window size is set to1.

The algorithms for MIMD-Poly and PIPD-Polyrepresented by the increase and decrease rules given by Eq. 3 and 8 are implemented as explained above.The window size and throughput of these algorithms are compared with the standard congestion algorithms for TCP[8,13,17].

TCP Friendliness of MIMD-Poly-a proof: We have used the TCP variants available inns-2 such as TCP/Tahoe, TCP/Reno, TCP/NewReno andTCP/Fast for comparison[18]. We now analyze the throughput of the polynomial algorithm as a function of the loss-rate it experiences. Weassume the MIMD-Poly algorithm. The analysis is similar tothe analysis explained in[8,16]. Using the Increase rule of MIMD-Poly algorithm and by linear interpolation of windowsize between Wt and Wt+R we get:

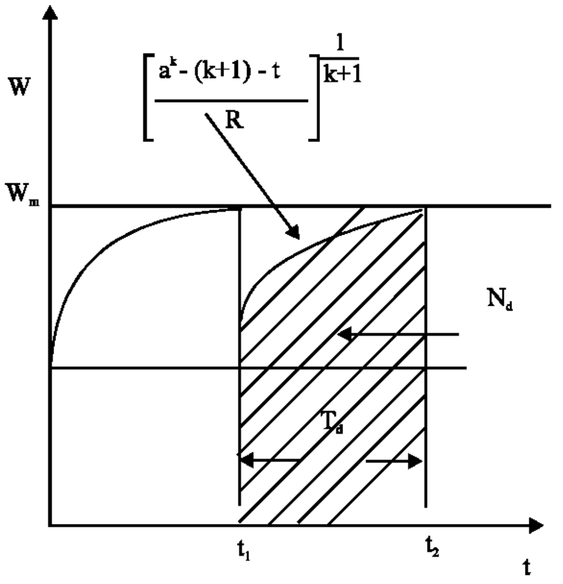

Fig. 1: Functional form of window vs. time curve

$$\frac{dW}{dt}=\frac{\alpha^k}{W^k.R} \qquad (11)$$

$$\frac{W^{k+1}}{k+1}=\frac{\alpha^k.t}{R}+C \qquad (12)$$

where C is the constant of integration.

The functional form of this curve is shown in Fig. 1.We are interested in the two parameters T_d and N_d marked in the Fig. 1; where T_d is the time between two successive packet drops and N_d is the number of packets received between two successive drops.

Let Wm be the maximum value of the window W_t attime t_2, at which congestion occurs. Then the expressions for T_d and N_d are evaluated as given below:

$$T_d = t_2 - t_1$$

$$T_d=\frac{R}{\alpha^k(k+1)}\left[W^{K+1}-(W_m-\beta^l W_m^l)^{k+1}\right]$$

$$T_d=\frac{R.W_m^{k+1}}{\alpha^k.(k+1)}\left[1-(1-\beta^l W^{l-1})^{k+1}\right] \qquad (13)$$

$$T_d=\frac{R.W_m^{k+1}}{\alpha^k.(k+1)}(\beta^l.W_m^{l-1})^{k+1}$$

$$T_d=\frac{\beta^l.R}{\alpha^k.(k+1)}W_m^{k+1}$$

N_d is the shadedarea under the curve in Fig. 1.

$$N_d=\frac{(k+1)^{\frac{1}{k+1}}}{R}\int\left[\frac{\alpha^k.t}{R}\right]^{\frac{1}{k+1}}dt \qquad (14)$$

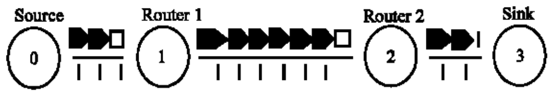

Fig. 2: Topology to simulate the packet drops

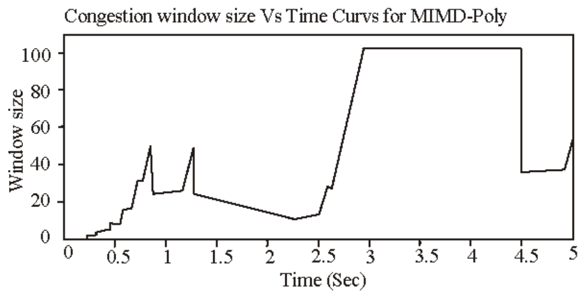

Fig. 3: Window size variation of MIMD-Poly Algorithm for the simulated packet drops

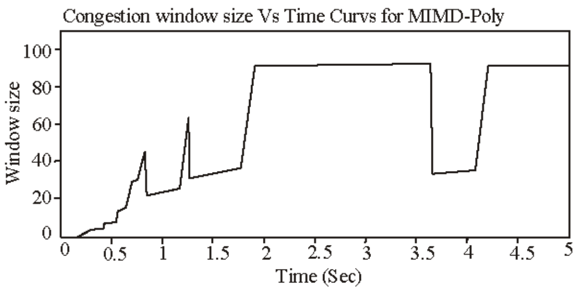

Fig. 4: Window size variation of PIPD-Poly Algorithm for thesimulated packet drops

Calculating the integral, we get

$$N_d=\frac{\beta^l}{\alpha^k}W_m^{k+l+1} \qquad (15)$$

The average throughput, λ of a flow using the polynomial congestion control, MIMD-Poly is the number of packets sent between successive drops (N_d) divided by theduration between drops (T_d). The packet loss probability p = 1/N_d. Writing λ and p in terms of Wm by substituting the expressions for N_d and Td yield :

$$\lambda \propto 1/p^{1/k+l+1} \qquad (16)$$

This implies that for polynomial Model 1 to be TCP-compatible as per[16], λ must vary as 1/ p 0.5 and hence k+l= 1[8,20]. This analysis can be extended to find the throughput of the PIPD polynomial algorithm also. Were strict with the MIMD-Poly and in future we will be attempting to prove the TCP-compatibility of the throughput relation for the PIPD and other polynomial models.

Simulation of Packet Drops: The variations of window size with time for theMIMD-Poly and PIPD-Poly algorithms are studied by usingthe ns-2 simulations. The network topology, consisting of asource node (n0) connected to a receiver node (n3) through two routers (n1 and n2) as shown in Fig. 2 is used foranalyzing the performance of the algorithms during packet drops.

We have attached a TCP agent to the source node and a TCP/Sink to the receiver node. The links are assumed to bebidirectional with a bandwidth of 1Mbps. The link connectingthe two routers is assumed with a buffer capacity of 20 and a Drop Tail buffer management algorithm. The data packets generated by an FTP Source are connected to the source node.The TCP agents are implemented with the MIMD-Poly and PIPD-Poly increase and decrease rules. These rulesare implemented into the TCP agent program tcp.cc. The twoalgorithms are selected by choosing the values for theTCP/Agent variable window Option. If window Option is 9,then MIMD-Poly is chosen and if it is 10, then PIPD-Poly ischosen. We have simulated the selective packet drops and theeffects of these drops on window size variations are recorded.The Fig. 3 and 4 show the window size vs. time plot for thesimulated packet drops for MIMD-Poly and PIPD-Poly. Thepackets with sequence numbers 50,150 and 700 are dropped.In MIMD-Poly algorithm the packets are dropped at the timeinstances T1 = 0.78s, T 2 = 1.205s and T3 = 4.428s. In PIPD-Poly algorithm the packets are dropped at the time instances T1 = 0.78s, T 2 = 1.205s and T3 = 3.577s. The plots given in Fig. 3 and 4 shows that the two algorithms use the increaseand decrease rules given by Eq. 3 and 8. The windowsize of the PIPD-Poly algorithm increases faster than theMIMD-Poly algorithm during the congestion avoidance phase.

PERFORMANCE EVALUATION OF POLYALGORITHMS IN WIRED TCP NETWORKS

we present the results of our ns-2simulation of various polynomial algorithms in wired TCPnetworks[18]. We start by investigating the connections running the TCP-compatible polynomial algorithms MIMD-Poly and PIPD-poly represented by Eq. 3 and 8. Thesimulations use the topology shown in Fig. 5

It consists of 6 connections sharing a bottleneck linkwith total bandwidth equal to b, where all connections have analmost identical round-trip propagation delay equal to RTT. Each polynomial flow uses a modified TCP with AIMD algorithm replaced by the polynomial family; other mechanisms like slow-start and time-out remain unchanged asalready mentioned in section 4 of this paper. Each source always has data to send, modeled using ns's FTP application. We have conducted the simulation and observedthe values of window size and throughput. The figures andgraphs display this information.

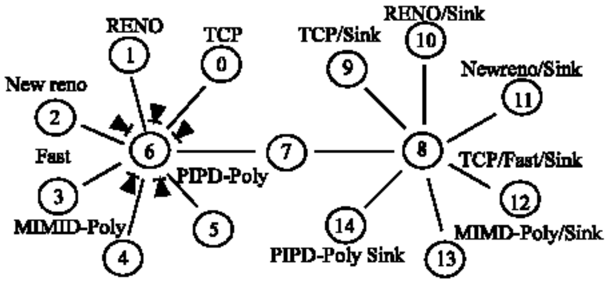

Fig 5: Dumbbell topology used for wired TCP network

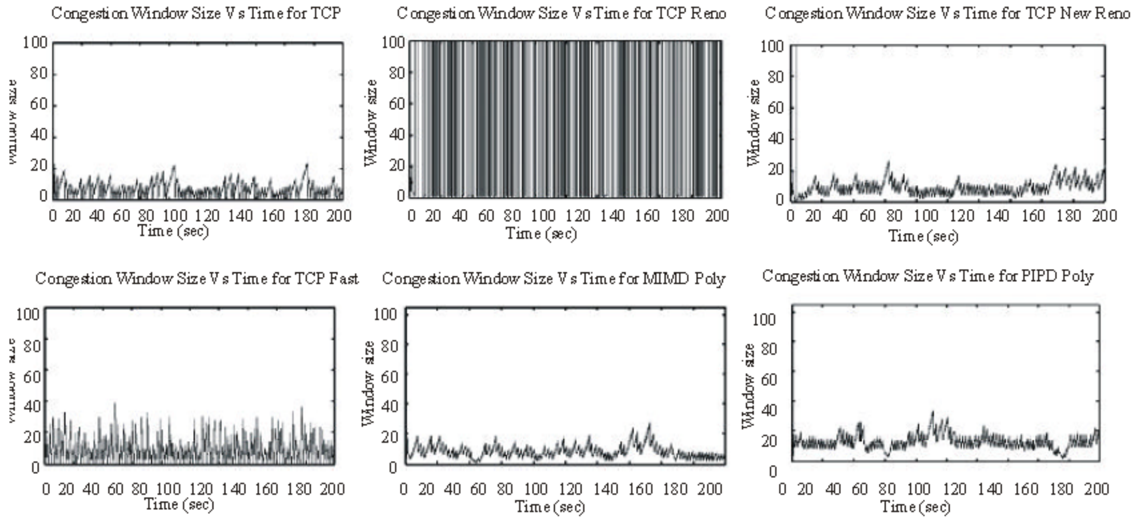

Fig. 6: Window size vs. time of TCP, TCP/Reno, TCP/NewReno, TCP/Fast, MIMD-Poly and (d) PIPD-Poly. (α = 1.75 and β=0.1)

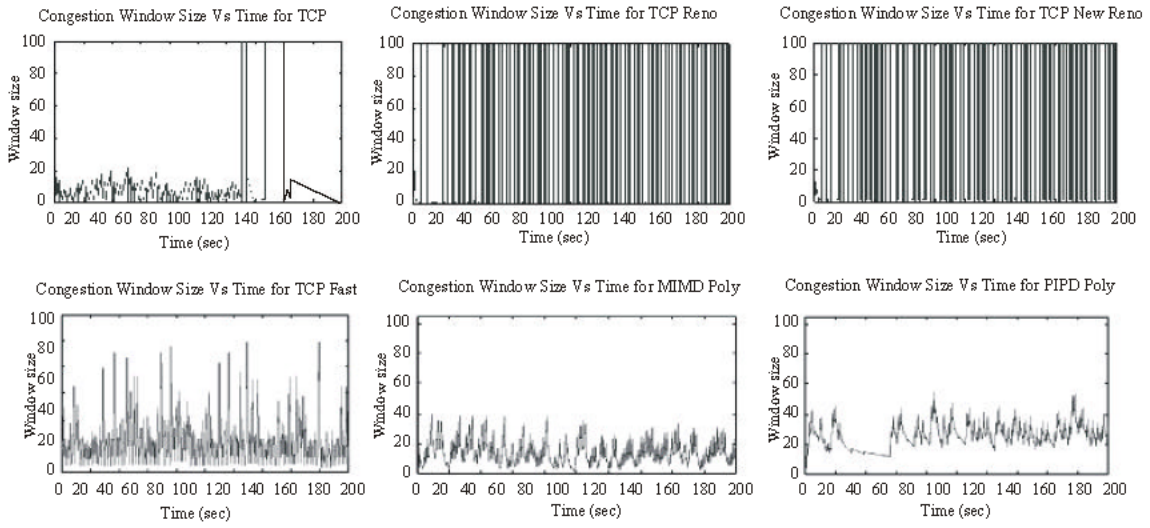

Fig. 7: Window size vs. time of TCP, TCP/Reno, TCP/NewReno, TCP/Fast, MIMD-Poly and (d) PIPD-Poly. ($\alpha = 0.33$ and $\beta = 0.2$)

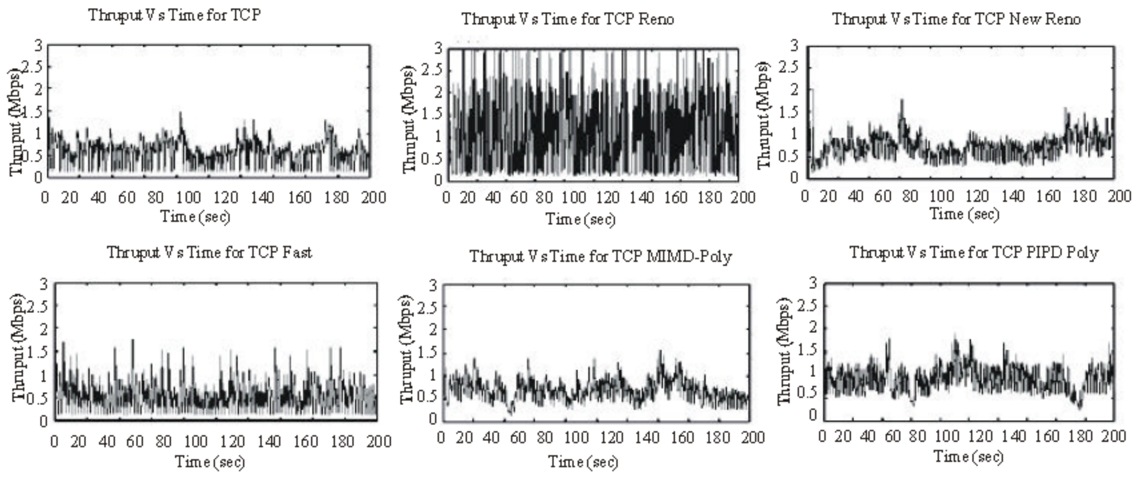

Fig. 8: Throughput vs. time of TCP, TCP/Reno, TCP/NewReno, TCP/Fast, MIMD-Poly and (d) PIPD-Poly. ($\alpha = 1.75$ and $\beta=0.1$)

The results are obtained using the Drop Tail buffermanagement algorithm at the bottleneck gateway[16,17]. Figure 6 and 7 show the variation of window size over asimulation period of 200 sec for TCP, TCP/Reno,TCP/NewReno, TCP/Fast, MIMD-Poly and PIPD-Polymodels. The results show that both congestion control algorithms are TCP-compatible and MIMD-Poly consumes more bandwidth than PIPD-Poly. Figure 8 and 9 show the throughput of the TCP,TCP/Reno, TCP/NewReno, TCP/Fast, MIMD-Poly and PIPD-Poly.The throughput (λ) is modeled using the Eq. 17.

$$\lambda = \frac{c.s}{R.\sqrt{p}} \quad (17)$$

where s is the segment size, R *is* the round trip time, p is thepacket loss rate and c is a constantvalue commonly approximated as$1.5\sqrt{2/3}$. The graphs show that the throughput of MIMD-Poly and PIPD-Poly are high compared with that of all the variants of TCP.

PERFORMANCE EVALUATION OF POLYALGORITHMS IN WIRED TCP NETWORKS WITHMULTICAST ROUTING PROTOCOLS.

Multicasting is very much essential to share the same information among a group of interconnected users. Distribution of real time audio and video to a group of hostswho have joined a distributed conference

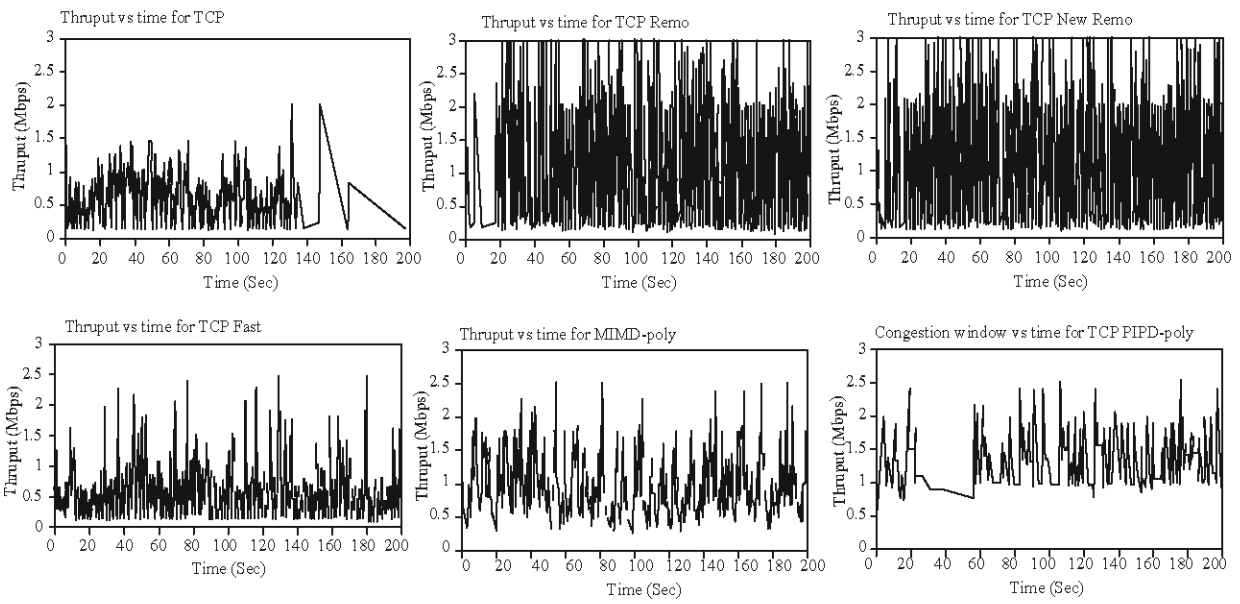

Fig. 9: Throughput vs. time of TCP, TCP/Reno, TCP/NewReno, TCP/Fast, MIMD-Poly and (d) PIPD-Poly. (α = 0.33 and β=0.2)

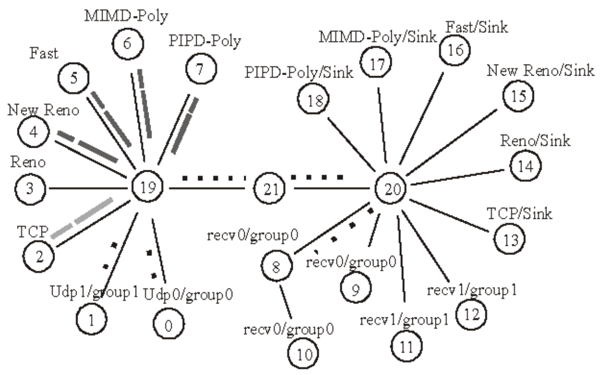

Fig. 10: Dumbbell topology used for multicast network

is an example of the application of multicasting[9]. This multicast traffic will besharing the network band width with other data traffic. Hence the protocols designed for data transfer should take care of such situations[23].

To examine the performance of the proposed algorithms MIMD-Poly and PIPD-Poly in interacting with the multicasting, we have simulated a network with dumb belltopology as shown in Fig. 10. Two multicasting groups areformed: one with node n0 as the source and nodes n8, n9 andn10 as receiving members forming group0, the other withnode n1 as the source and n11 and n12 as receiving members forming group1. The receiving members can join or leave the group at any time. The simulation scenario for the members to join and leave is randomly chosen.

For the multicast groups we have attached the CBR traffic to the source nodes. To study the interaction of the TCP traffic, agents of the TCP variants such as Tahoe, Reno, New Reno and Fast along with the proposed Poly algorithms are attached to these sources and FTP data traffic is sent throughthese source nodes. These data traffic contend to share the bandwidth of the bottleneck link with the multicast traffic. The simulation results of the TCP sources such as window size and throughput are recorded.

Two multicast routing strategies are assumed for thesimulations. One is the Centralized multicast mode and theother is the Dense Mode. In Dense mode, two variations suchas pim-dm and dvmrp are used. The main difference between theses two variations is that dvmrp maintains parent-child relationships among nodes to reduce the number of links overwhich data packets are broadcast[18].

The results of the simulations i.e. the window size vs.time graphs for all the TCP sources are plotted. The Fig. 11 and 12 show the window size variations of TCP variants Tahoe, Reno, New Reno, Fast, MIMD-Poly and PIPD-Polywith the two multicast routing strategies Ctrm and DM. The results show that the two Proposed Algorithms behave similarto other TCP variants with an improvement in total through put.

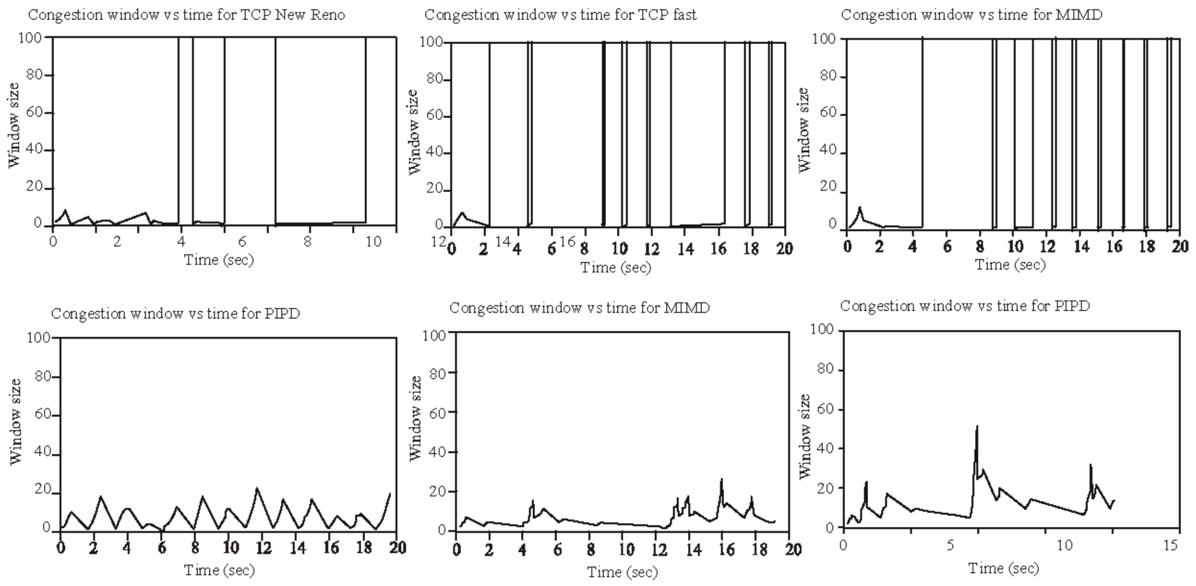

Fig. 11: Window size vs. time of TCP, TCP/Reno, TCP/NewReno, TCP/Fast, MIMD-Poly and PIPD-Poly. (Multicast Centralized mode)

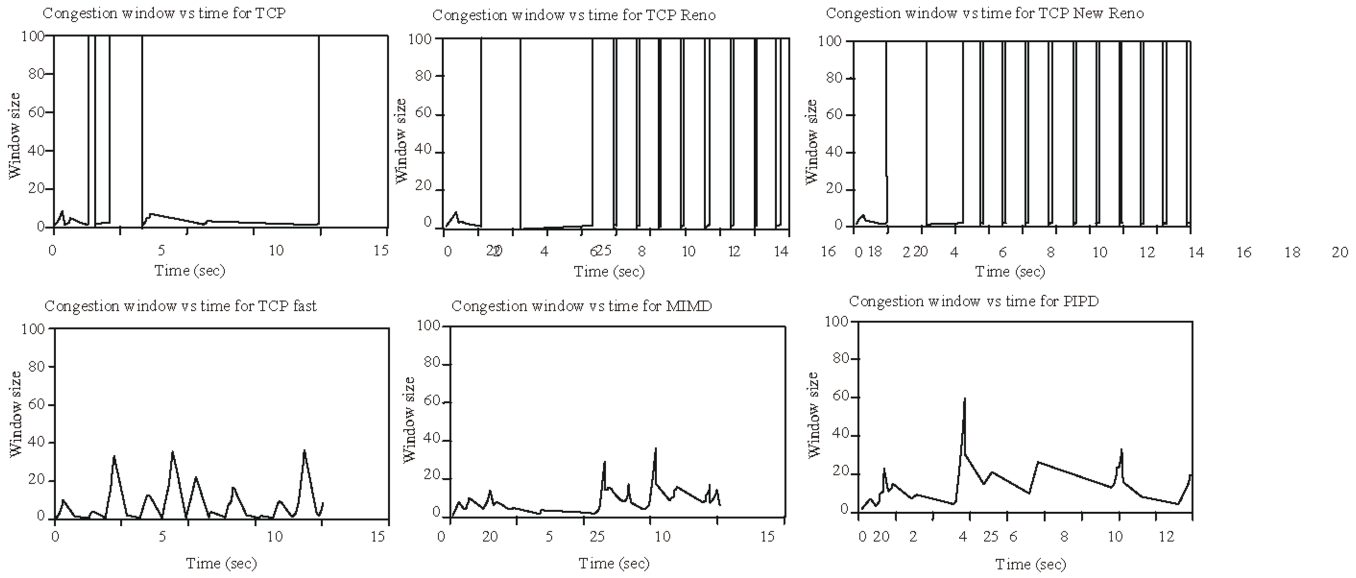

Fig. 12: Window size vs. time of TCP, TCP/Reno, TCP/NewReno, TCP/Fast, MIMD-Poly and PIPD-Poly. (Multicast Dense mode)

PERFORMANCE EVALUATION OF POLYALGORITHMS IN MOBILE AD HOC NETWORKS

A Mobile Ad Hoc Network is a network in which agroup of mobile computing devices communicate amongthemselves using wireless radios, without the aid of a fixed networking infrastructure[23]. They can be used anywhere thata fixed infrastructure does not exist, or is not desirable. MobileAd Hoc networks provide an extension to the Internet. Since TCP/IP is the standard network protocol on the Internet, itsuse over mobile Ad Hoc network is a certainty[11,24]. Herewe analyze the performance of the proposed window based Polynomial congestion control algorithms over the mobile Ad Hoc networks and compare them with the

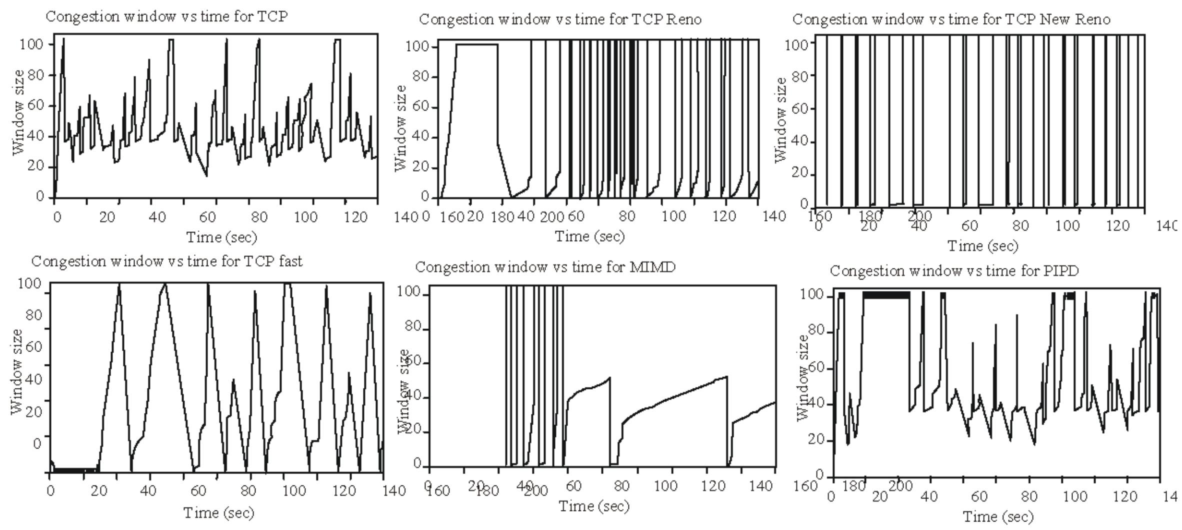

Fig. 13: Window size vs. time of TCP, TCP/Reno, TCP/New Reno, TCP/Fast, MIMD-Poly and PIPD-Poly. (Mobile Ad Hoc AODV)

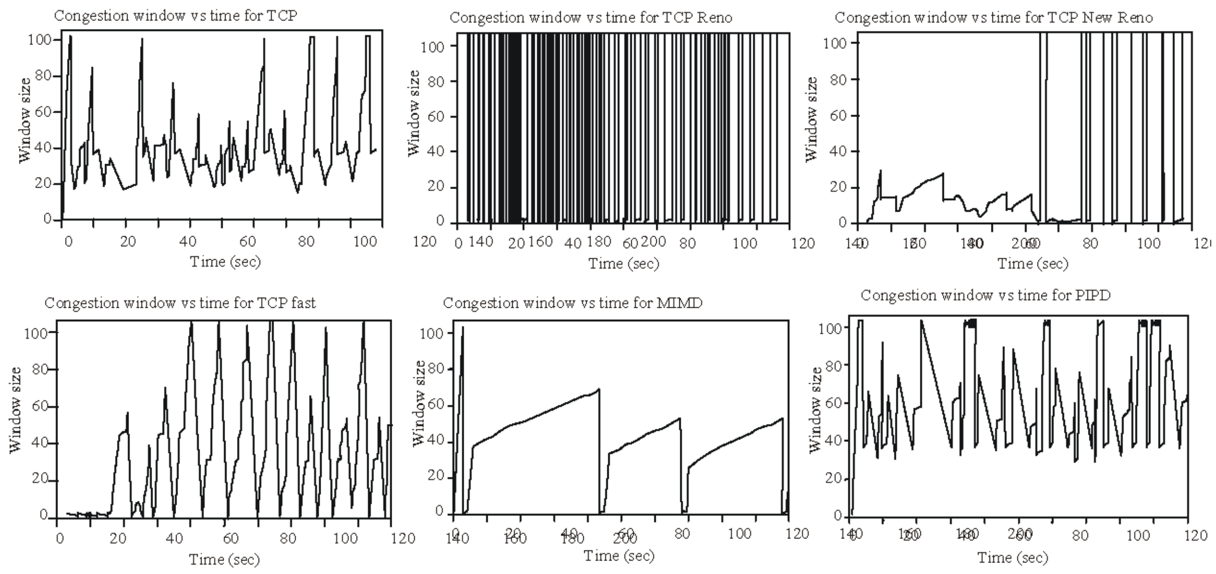

Fig. 14: Window size vs. time of TCP, TCP/Reno, TCP/NewReno, TCP/Fast, MIMD-Poly and PIPD-Poly. (Mobile Ad Hoc DSDV)

standard algorithms implemented for TCP, TCP/Reno, TCP/New Reno andTCP/Fast[13,17].

We use an initial simulation topology containing 12 nodes. We simulate a mobility scenario, which creates arandom movement to the nodes. Figure 13 shows twosnapshots of the topology we used at different simulation instants. It shows the mobility simulated with different sources delivering the TCP packets.

In our simulations we have assumed the followingrouting algorithms used for Mobile Ad Hoc networks: DynamicSource Routing (DSR), Dynamic Destination Sequenced Distance Vector (DSDV) Routing and Ad-hoc On-demandand Distance Vector (AODV) Routing[18,20]. The sametopology and mobility scenario is assumed for each routing algorithms.

In each case we have simulated the window based polynomial congestion control algorithms. The performance of the two models MIMD-Poly and PIPD-Poly are recorded along with the sources implemented with standard TCP, TCP/Reno,

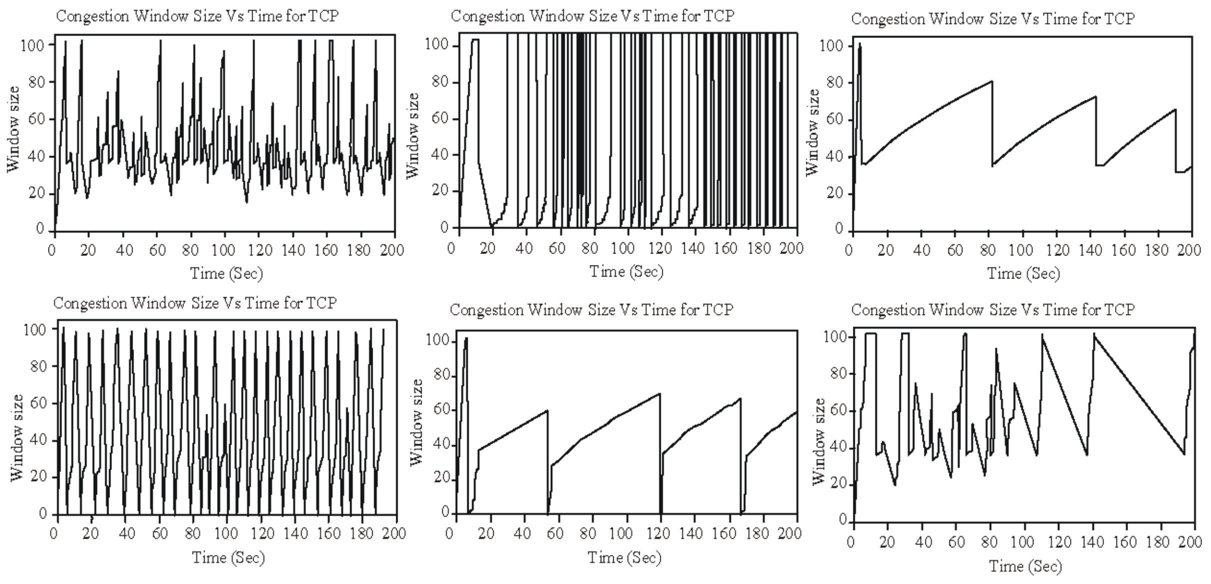

Fig. 15: Window size vs. time of TCP, TCP/Reno, TCP/NewReno, TCP/Fast, MIMD-Poly and PIPD-Poly. (Mobile Ad Hoc DSR)

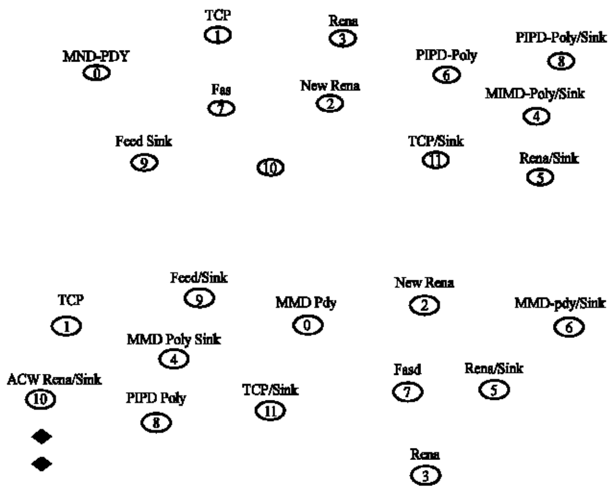

Fig. 16: Snapshots of mobile Ad Hoc topology at two different instantsof time.

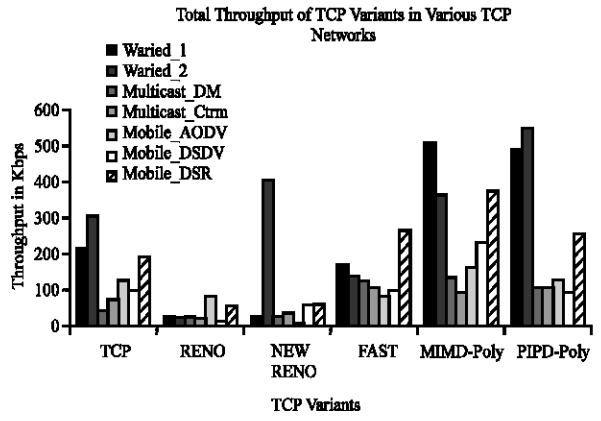

Fig. 17: Total throughput of TCP Variants

TCP/NewReno and TCP/Fast. Fig. 13, 14 and 15 display the window size adjustment using these algorithms.The throughput is evaluated based on the Eq. 17.

The rz esults show that the PIPD-Poly and MIMD-Poly algorithms improved the total throughput regardless of the routing method used. Figure 17 shows the comparison of the total throughput of TCP, TCP/Reno and MIMD and PIPD congestion control for wired and mobile Ad Hoc networks.

SCALABILITY AND FAIRNESS

We have tested the scalability of the Poly algorithms.The numbers of nodes are increased to 26 and the dumb belltopology shown in Fig. 5 is extended to the to pology containing 26 source nodes and 26 receiver nodes connected by a single bottleneck link through two routers. The topologyis shown in Fig. 18. The sources are randomly chosen to have one of TCP variants and the two proposed algorithms MIMD-Poly and PIPD-Poly. The throughput of each link is evaluated and found that the proposed algorithms

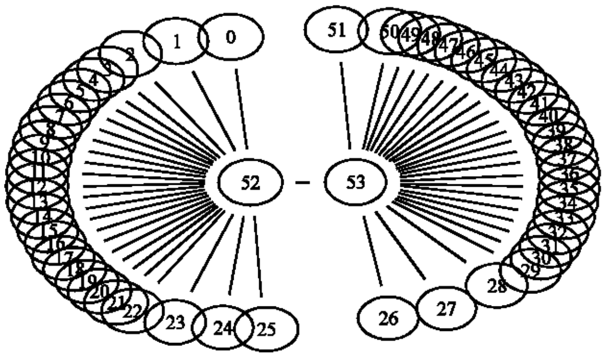

Fig. 18: Topology for testing the scalability

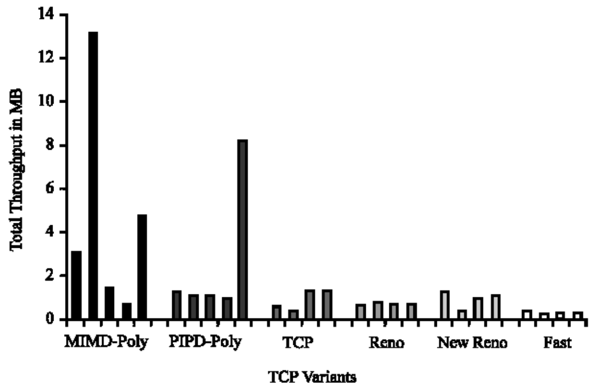

Fig. 19: Total throughput of TCP Variants with 26 sources

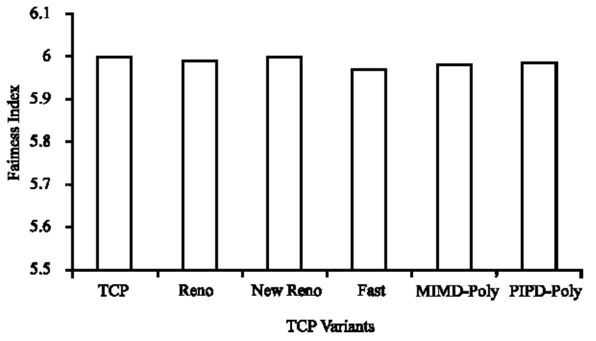

Fig. 20. Fairness Index of TCP Variants

perform better by acquiring the bandwidth aggressively. The Fig. 19 shows the total throughput of individual links for the assumed configuration.

For the wired TCP network shown in figure 5, wehave calculated the fairness index using the following model. The fairness index f (λ)_is given by[23]

$$f(\lambda) = \frac{\left[\sum_{i=1}^{n} \lambda_i\right]^2}{\sum_{i=1}^{n} \lambda_i^2}$$

All the sources are assumed to be of the same typeand the fairness index is evaluated. The values of the various through puts and the corresponding fairness index are shown in the Fig. 20. The Fairness index for an n node network, contending for the bandwidth through the bottle neck link is tobe approximately n. Hence the value should be 6 for the topology shown in figure. All the TCP variants and the two proposed models show the fairness index as shown in the Fig. 19.

CONCLUSIONS

In this study, we presented and evaluated a family of nonlinear congestion control algorithms, called polynomial algorithms. We have considered the MIMD-Poly and PIPD-Poly models of the polynomial family for our experimentation.These polynomial algorithms generalize the familiar class of linear algorithms. We showed that for one of the restricted model of the polynomial family, MIMD-Poly, the throughputλ_μ1/p 1/k+l+1, where p is the packet loss rate it encounters.This shows that the model is TCP compatible.

Our simulation results showed good performance and interactions between polynomial algorithms (MIMD-Poly andPIPD-Poly) and TCP using standard algorithms in wired andmobile Ad Hoc networks. The algorithms MIMD-Poly andPIPD-Poly obtain higher long-term throughput than the standard algorithms for TCP, TCP/Reno, TCP/NewReno andTCP/Fast. The Fig. 6, 7, 11, 12, 13, 14 and 15 show the variation of congestion window with respect to time for theTCP, TCP/Reno, TCP/NewReno, TCP/Fast, TCP/MIMD andTCP/PIPD algorithms in the case of wired network (withunicast and multicast) to pology shown in Fig. 5 and 10 andin the case of Mobile Ad Hoc networks. Fig. 17 displays thetotal throughput for the various TCP variants in wired andmobile Ad Hoc networks. The topology of the mobile Ad Hocnetwork is shown in Fig. 16.

We believe the results presented in this paper lead toan understanding of the issues involved in the increase anddecrease phases of a congestion control algorithms.

REFERENCES

1. Van, J., 1988. Congestion Avoidance and Control. ACM Computer Communication Review, Proceedings of the Sigcomm '88 Symposium in Stanford, CA, 18: 314-329.
2. Widmer, J., R. Denda and M. Mauve, 2001. A Survey on TCP-FriendlyCongestion Control. IEEE Network Magazine.

3. Congestion Control Schemes for TCP/IP Networks currentimplementation in BSD Unix. Douglas E. Comer, Internetworking withTCP/IP Volume I, Englewood Cliffs, New Jersey, Prentice Hall, 1991.
4. Stevens, W.R., 1994. TCP/IP Illustrated, Vol. 1, Addison-Wesley, Reading, MA,
5. Richard, Y. and S.S. Lam, 2000. General AIMD congestioncontrol. In: Proceedings of ICNP.
6. Sally, F. and K. Fall, 1999. Promoting the use of end-to-end congestioncontrol in the Internet. IEEE/ACM Transactions on Networking, 7: 458-472.
7. Rejaie, R., M. Handley and D. Estrin, 1999. RAP: An End-to-end Rate-based Congestion Control Mechanism for Real time Streams in the Internet, inProc. IEEE INFOCOM, 3: 1337-1345.
8. Shudong, J., L. Guo, I. Matta and A. Bestavros, 2001. Aspectrum of TCP-friendly window-based congestion control algorithms. Tech. Rep. BU-CS-2001-015, Computer Science Department, BostonUniversity, Available at http:// www. cs.bu. edu/techreports/2001-015-spectrum-tcp-friendly.ps.Z.
9. K. Seada, S. Gupta and A. Helmy, 2002. Systematic Evaluation of MulticastCongestion Control Mechanisms. SCS SPECTS.
10. K. Sundaresan, V. Anantharaman, H.Y. Hsieh and R. Sivakumar, 2003. ATP: Areliable Transport Protocol for Ad Hoc Networks, ACM Mobihoc.
11. Holland, G. and N.H. Vaidya, 1999. Analysis of TCP Performance over MobileAd Hoc Networks. ACM MOBICOM ¶99, Seattle.
12. Georgi, K., 2005. A Simulation Analysis of the TCP ControlAlgorithms. International Conference on Computer Systems and Technologies-CompSys Tech.
13. Kevin, F. and S. Floyd, 1996. Simulation-based Comparisons of Tahoe, Reno and SACK TCP. Computer Communications Review
14. Zheng, W. and J. Crowcroft, 1991. A New Congestion Control Scheme: Slow Start and Search (Tri-S). ACM Computer Communication Review, 21: 32-43.
15. Lawrence, S.B. and S.W.O. Malley, 1994. TCP Vegas: New Techniques for Congestion Detection and Avoidance, in SIGCOMM '94 Conference on Communications Architectures and Protocols, (London,United Kingdom), pp: 24-35.
16. Deepak, B. and H. Balakrishnan, 2001. "Binomial congestion controlalgorithms, in Proceedings of IEEE INFOCOM.
17. Bogdan Moraru, Flavius Copaciu, Gabriel Lazar and Virgil Dobrota,³Practical Analysis of TCP Implementations: Tahoe, Reno, NewReno", Technical University of Cluj-Napoca
18. "ns-2 Network Simulator," http:// www.isi. edu/ nsnam/ns/, 2000.
19. Sally, F., 1995. TCP and Explicit Congestion Notification. ACM Computer Communication Review, 24: 8-23.
20. Bansal, D. and H. Balakrishnan, 2000. TCP-friendly Congestion Control forReal-time Streaming Applications. Tech. Rep. MIT-LCS-TR-806, MITLaboratory for Computer Science
21. Floyd, S. and V. Jacobson, 1993. Random Early Detection Gateways for Congestion Avoidance. IEEE/ACM Transactions on Networking, 1: 4
22. Tejas, K. and V. Kakadia described Analysis of Congestion Control StrategiesFor TCP Variants Using Droptail and RED Queuing Disciplines Departmentof Computer Science University of Southern California, Los Angeles.
23. K. Seada and A. Helmy, 2002. Fairness Evaluation Experiments for MulticastCongestion Control Protocols. Technical Report 02-757,University ofSouthern California, CS Department.
24. Chen, K., Y. Xue and K. Nahstedt, 2003. On setting TCP's congestion windowlimit in mobile Ad Hoc networks. IEEE ICC ¶03, Anchorage, Alaska