Performance Analysis of Proactive and Reactive Transport Protocols using a Proactive Congestion Avoidance Model

Mohanchur Sarkar
SATCOM Application Area
Space Applications Centre, ISRO
Ahmedabad, India

K.K.Shukla
Department of Computer Engineering
Institute of Technology, BHU
Varanasi, India

K.S.Dasgupta
SATCOM Application Area
Space Applications Centre, ISRO
Ahmedabad, India

ABSTRACT

This paper attempts to establish the fact that Proactive transport protocols are better than conventional Reactive transport protocols in terms of overall utilization of resources. This has been proved analytically in this paper using a mathematical model developed for proactive congestion avoidance. The paper also provides a mathematical framework for Proactive Transport Protocol and predicts the congestion window evolution and throughput achieved by single and multiple proactive TCP flows. The congestion window evolution and throughput of a Proactive TCP connection are modeled as functions of the bandwidth, delay, packet size, number of concurrent TCP flows and the penalty factor used by the proactive protocols. Results obtained from the model are verified with ns2 simulation. The optimal value of the penalty factor is derived mathematically from the designed model. It has also been shown how the derived optimal values enhances the performance of the protocol and almost a near maximum utilization of resources is achieved by extensive simulation in ns2 considering both single and multiple TCP flows.

General Terms

Algorithms, Performance, Design, Experimentation, Verification.

Keywords

GEO, ns2, Proactive TCP, SACK, Vegas, Multiple Flow Model

1. INTRODUCTION

The TCP Protocols can be broadly classified into two categories, reactive protocols and proactive protocols. The reactive protocols do not take any action unless and until the problem really happens. The congestion window is allowed to grow as long as the acknowledgements return, signaling allowable capacity in the network till the point when Duplicate ACKs start coming signifying a loss of packet due to congestion or channel error. At that point corrective actions are taken mainly by reducing the congestion window and slow start threshold by different amounts with the intent to allow the network to come out of the congested state. The reactive protocols with a view to maximize the throughput always drives the network to the maximum capacity after which every connections suffer the collateral damage caused by the overestimation of the channel capacity. All the AIMD TCP protocol variants like Tahoe, Reno, NewReno, SACK[7], FACK[10], Peach[3], Peach+[4], TP-Planet[5] fall in this category. Reactive algorithms try to solve the problem without considering the root cause of the problem. On the other hand the proactive protocol tries to anticipate the overestimation of the network capacity and start taking corrective action to avoid the incipient congestive meltdown of the network.

The main problem with the reactive protocols is that the congestion window of all the flows participating in the network goes on increasing till the point that the buffer capacity in the routers are overflowed and all flows experience the congestion originated packet loss. At this time the congestion window of all the flows are reduced to half and through fast retransmit and congestion avoidance the network comes out of the congested state. The point is that with an aim to maximize the capacity available, the network is periodically driven to its maximum capacity and then all the flows suffer the collateral damage created by some of the over ambitious flows. If on the other hand protocols can be designed which can priory sense an incoming congestion from some parameters of the connection then corrective actions can be taken before the network moves to its maximum allowable capacity. The corrective action can be a decrease of the congestion window with the signal of an incipient congestion. The amount by which the reduction will be done can be termed as the penalty factor which is multiplied with the prevailing value of the congestion window once the incipient congestion is detected.

In this paper we have examined the range of values for the penalty factor so that the proactive approach toward congestion control can be beneficial to the overall capacity utilization of the network. The reactive protocols are straight forward in the sense that they go on increasing their utilization to the point that there is no capacity left in the network. The implementation is simple and the utilization is obvious but the collateral damage the connections experience at the time of a congestive meltdown has to be determined. On the other hand the design of a proactive protocol is challenging in terms of estimating the incipient congestion. The proactive protocols should employ a much cooperative approach in dealing with capacity utilization and the network should be operated below its maximum capacity and some room should be kept for uncontrollable flows which can any time clog some capacity of the network. The approach of the protocol should be more oriented toward achieving an overall maximization of throughput for all the connections prevailing in the network rather than just maximizing the flow of one connection. It’s a question of giving importance to the social self than on the selfish self.

2. MULTIPLE FLOW MODEL

In this section we will try to show how proactive protocols can lead to a lesser underutilization of channel capacity in comparison to reactive protocols. To establish this fact let us consider a model of the network similar to the proposals in [2][9].

Our model is based on the TCP congestion avoidance algorithm and the network topology illustrated in Fig1 [2]. In the network configuration, N TCP flows share one bottleneck link L from router G0 to G1. The buffer size of the router G0 is Q (packets). G0 performs first-in-first-serve and drop-tail queuing. The bandwidth of L is B (bits/second). During congestion avoidance, each of the TCP sources increases the congestion window for each acknowledgment it receives. Consequently, in each round-trip time (rtt) period, the TCP source increases its congestion window by one packet size S. We use w_i to indicate the window of the ith TCP flow.

$$w_i(RTT_{j+1}) - w_i(RTT_j) = 1 \quad (1)$$

Where, w_i (RTT_j) denotes the value of the congestion window in terms of packets at jth RTT from the start of connection. This shows that the congestion window increases by 1 every RTT.

For each TCP connection, the rtt value is expressed as

$$rtt = T_q + T_p + T_s \quad (2)$$

where, T_q is the queuing delay, T_p is the propagation delay and T_s the service delay. Since L is the only bottleneck link, on the forward link the packets will only queue up at the router G0. So we have

$$T_q = \frac{l}{B} \quad (3)$$

where, l is the queue length at the router G0 and B is the bandwidth of L. There are N concurrent TCP flows. Originating from S_i, the ith TCP flow travels over link Li to router G0, link L to router G1, and link K_i to reach the destination R_i. The link L between G0 and G1 is the bottleneck link. All links are symmetric and full-duplex. The router G0 is a drop-tail router with buffer size Q. We have assumed that Tp and Ts are all constants. Here, we use $T = T_p + T_S$ (4)

Combining (2), (3), and (4), we get the following equation:

$$rtt = \frac{l}{B} + T \quad (5)$$

As the TCP senders increase their window, the packets sent out will first fill the bit pipe between G0 and G1, then the buffer of G0. The capacity of the pipe is the product of B and T, i.e., BT in terms of bits and BT/S when expressed in terms of packets. The queue length l (packets) at G0 can be expressed as follows:

$$l = 0 \qquad \text{if } \sum_{i=1}^{N} w_i < \frac{BT}{S} \quad (6a)$$

$$l = \sum_{i=1}^{N} w_i - \frac{BT}{S} \qquad \text{if } \sum_{i=1}^{N} w_i > \frac{BT}{S} \quad (6b)$$

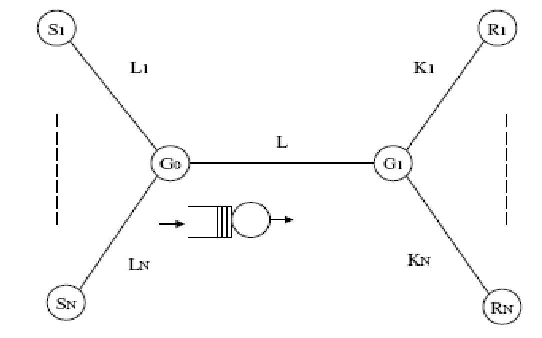

Fig1. Network Topology

As l increases beyond Q, the G0 buffer overflows and congestion occurs. We assume when l reaches Q, i.e. $\sum_{i=1}^{N} w_i = BT/S + Q$ all the TCP senders can detect the congestion occurrence. Upon detecting the congestion, all the TCP senders halve their windows, and resume window increase until the next congestion occurs. Thus $\sum_{i=1}^{N} w_i$ will oscillate between the value BT/S + Q and $\frac{BT/S+Q}{2}$, That is,

$$\frac{BT/S+Q}{2} < \sum_{i=1}^{N} w_i < BT/S + Q \quad (7)$$

Because all TCP flows have the same propagation delay, after some time their window evolution will become synchronized [11], and the TCP flows will achieve fairness over the bottleneck link [12]. Therefore we can assume that the congestion window sizes of all the TCP sources are equal to each other at any time. We use w to indicate the common window size, and

$$\sum_{i=1}^{N} w_i = Nw \quad (8)$$

2.1 Performance comparison of Proactive and Reactive Transport Protocols

The sum of congestion window i.e. $\sum_{i=1}^{N} w_i$ for all the N concurrent flows which is equal to N*w can be seen to evolve as per Fig2 for a reactive protocol.

$$U_R = \frac{1}{2} * \frac{(\frac{BT}{S} + Q)}{2} * CE_R \quad (9)$$

where, U_R denotes the underutilization for reactive protocols which is equal to the area of the window formed as shown in Fig2. CE_R denotes the congestion epoch of reactive protocols which is defined as the number of RTTs after which the successive congestion events happen.

$$U_P = \sum_{i=1}^{PRf} \frac{1}{2}*(1-p)*(\frac{BT}{S}+Q)*PR_E)$$

$$= \frac{1}{2}*(1-p)*(\frac{BT}{S}+Q)*PR_E*PR_f \qquad (10)$$

where, Up denotes the underutilization for proactive Protocols and PR_E denoted the proactive reduction epoch which denotes the number of RTTs after which a proactive reduction happens. PR_f denotes the frequency of proactive reduction within the congestion epoch for Reactive protocols CE_R.. U_p is equal to the sum of area of all the small triangles shown in Fig3 with base PR_E and height (1-ρ) ((BT/S+Q).

$$U_R - U_P = \frac{1}{2}*(\frac{BT}{S}+Q)(\frac{CE_R}{2}-(1-p)*PR_E*PR_f) \qquad (11)$$

AS per definition $PR_f = \frac{CE_R}{PR_E}$, so

$$U_R - U_P = \frac{1}{2}*(\frac{BT}{S}+Q)(PR_E*PR_f)(p-\frac{1}{2}) \qquad (12)$$

Since BT/S > 0, Q > 0, PR_E > 0 and PR_f > 0

So, $U_R - U_P > 0$ if p > 0.5 which show that the underutilization in case of Reactive Protocols is more than Proactive protocols. Conversely it can be said that Proactive protocols leads to a better utilization of resources.

The penalty factor acts as a correction factor for the protocol. It has been shown that the value of penalty factor should be greater than 0.5 for the proactive approach to give better utilization. In the proactive approach the congestion window is made to oscillate below a point which marks as the region from where the probability of the network getting into congestion is more.

3. MATHEMATICAL MODEL OF PROACTIVE PROTOCOL

Authors have designed a Proactive Transport Protocol for performance enhancement of Satellite based networks [1]. The brief description of the protocol is given below. The protocol proposed in [1] is a Proactive protocol and it measures the mean RTT for every congestion window. If an increase in the mean RTT is experienced for three successive congestion window then it can be anticipated that the network is getting in to the congested state. Here the decision making criteria is that how many congestion window should be checked for successive increase to conclude an incipient congestion. If more number of windows is considered for deriving at the conclusion that the network is moving to a congested state, then the decision may be more accurate but it may lead to a point where the network gets so much congested that even taking corrective measures does not help. So it is decided that we take decision based on the mean RTT increase for three successive congestion windows. Out of the three windows the second window will detect an increase and gives us a direction and the next increase will confirm that detection so that corrective action can be taken. If we take a decision just by considering two windows then that may lead to false decisions.

After an incipient congestion is detected the congestion window is reduced by a penalty factor. The choice of the penalty factor is very important for the throughput of the protocol which means how much the window should be decreased so that the incipient congestion can be avoided and the network returns to a stable condition from the congested state. There is a tradeoff, if the penalty is too high then the throughput of the protocol will decrease and if it is too low, then the corrective action necessary for avoiding the congestion will not happen. A simulation has been carried out in ns2 [13] with different values of the penalty factor from 0.1 to 1 and 0.85 comes as the optimum value [1].

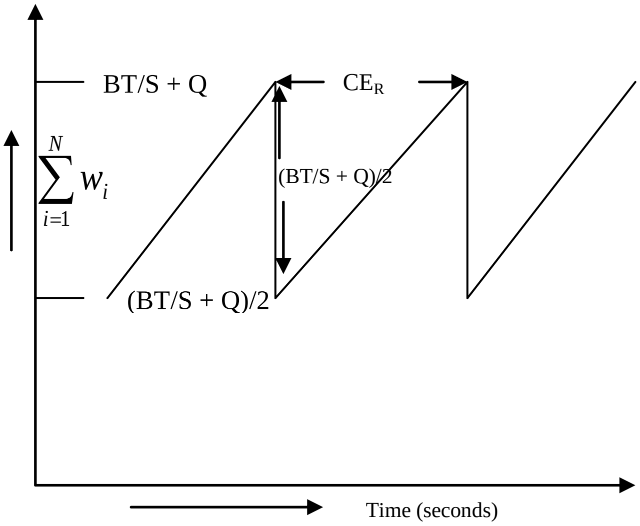

Fig2. Congestion Window Evolution for Reactive Protocols

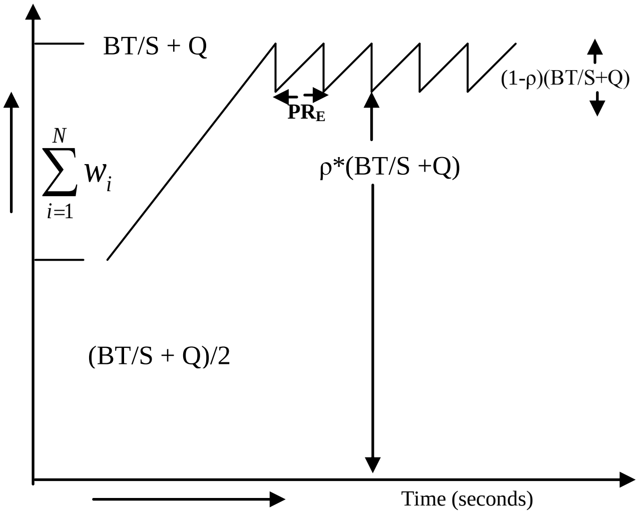

Fig3. Congestion Window Evolution for Proactive Protocols

3.1 Mathematical representation of Congestion Window Evolution

In this section we will try to formulate the way the congestion window of the proposed proactive protocol evolves. The congestion window of the proactive protocol moves through rounds of proactive reduction which lasts for PR_E RTTs before which three successive increases in mean RTT is signaled and a proactive reduction is called for. Here we are trying to formulate the value of the congestion window after successive proactive reduction epochs. If there are N connections sharing the resources and each is increasing its congestion window by 1 every RTT, then the overall increase in network load every RTT is N. Let the value of the sum of congestion windows for all n flows just before the proactive reduction is denoted by $cwnd_0$ which determines the overall load on the network. Within a Proactive Reduction Epoch at least it will take 3 RTT to signal a proactive reduction. This implies that at least 3N packets will be added to the overall network load. So the evolution of the congestion window can be though of a series of addition of network load every RTT and periodic proactive reduction. It k denotes the proactive reduction index which signifies the sequence of proactive reductions, then the value of the congestion window can be found as shown in Fig4. In Fig4 PRI denotes the Proactive Reduction Index.

PRI	Congestion window value
1	$p_*cwnd_0 + 3N$
2	$p(p_*cwnd_0 + 3N) = p^2{}_*cwnd0 + p_*3N + 3N$
3	$p(p^2cwnd_0 + p_*3N + 3N)= p^3{}_*cwnd0 + p^2{}_*3N + p_*3N + 3N$
4	...
k	$P^k{}_*cwnd0 + p^{k-1}{}_* 3N + p^{k-2}{}_* 3N + ... + p_*3N + 3N$

Fig4. Congestion window Evolution with proactive reduction epoch

So after kth proactive reduction the cwnd is given by
$cwnd (k) = p^k{}_*cwnd_0 + 3N_* \{pk^{-1}+ p^{k-2} + ... + p + 1\}$

$$= p^k{}_* cwnd_0 + 3N_* \frac{(1-p^k)}{(1-p)} \qquad (13)$$

It can be observed that during every proactive reduction epoch, an amount of packets given by (1-p) * cwnd is reduced from the overall system load and an amount of 3*N packets are added to the system load progressively with N packets every RTT.

Equation (13) is a special case of a generalized equation of congestion window evolution of proactive protocols, where $cwnd_0$ denotes the overall system load which triggers a proactive reduction and the addition of 3N packets implies the product of proactive reduction epoch PR_E with the number of concurrent flows in the network N. In this case the PR_E or the number of RTTs needed for successive trigger of proactive reduction is considered to be 3 which is the minimum number of RTTs needed to trigger a proactive reduction. This is because the logic that the proactive reduction is initiated with 3 successive increase of mean RTT.

So in a generalized way it can be thought that from overall network load $(1-p)*cwnd_0$ packets are decreased and $(N * PR_E)$ number of packets are added during every proactive reduction epoch. Now let us consider the specific case of the Proactive protocol proposed by the authors [1] where,

$$cwnd0 = \frac{B*T}{S} + 3 \text{ and } PR_E = \frac{(1-p)*(\frac{BT}{S}+3)}{N}$$

So the overall load on the network as given by, Cwnd (k) =

$$p^k * (\frac{B*T}{S} + 3) + [\ \frac{(1-p)*(\frac{BT}{S}+3)}{N} * N] * \frac{(1-p^k)}{(1-p)} \qquad (14)$$

This gives a value of $\frac{B*T}{S}+3$ for all k. This is what happens in the proactive protocol. As long as the load on the network is less than B*T/S there will be no queue developed as per (6a) so there will be no increase in the mean RTT only every RTT, N packets will be added to the overall network load.

When the load on the network crosses B*T/S, queue will start to build up as per (6b) and there will be an increase in the experienced RTT. With three successive increase in mean RTT a proactive reduction will be called for which reduces the value of congestion window.

3.2 Computation of average throughput

For the calculation of the average throughput we have to concentrate on one proactive reduction epoch. From (14) it can be seen that every time a proactive reduction is initiated when the overall network load is $\frac{B*T}{S}+3$, which may be seen as the level of network load which the protocol tries to maintain. Immediately after the proactive reduction is performed the value of $\sum_{i=1}^{N} cwnd_i$ is given by

$$cwnd_{initial} = p * (\frac{B*T}{S} + 3) \qquad (15)$$

and the value of $\sum_{i=1}^{N} cwnd_i$ just before the proactive reduction is given by

$$cwnd_{final} = \frac{B*T}{S} + 3 \qquad (16)$$

So, $cwnd_{average} = (cwnd_{initial} + cwnd_{fianl}) /2$

$$cwnd_{average} = \frac{1}{2} * \{\ p * (\frac{B*T}{S} + 3) + \frac{B*T}{S} + 3 \}$$

$$= \frac{(1+p)*(B*T+3S)}{2S} \qquad (17)$$

This signifies the average number of packets sent through the network within a proactive reduction epoch shared by N connections. In the steady state the available network resources can be through of being equally shared as per (8) so the individual congestion window for the ith flow is

$$cwnd_i = \frac{1}{N} * \frac{(1+p)*(B*T+3S)}{2S} \qquad (18)$$

To arrive at an average throughput calculation for the protocol we have to calculate the average value of RTT. To calculate the average RTT value we first need to calculate the exact duration of the Proactive Reduction Epoch which denotes the time in which $\frac{(1+p)*(B*T+3S)}{2S}$ packets are sent through the network.

Whenever a proactive reduction happens the overall load on the network will be reduced from $\frac{B*T}{S}+3$ packets to p * ($\frac{B*T}{S}+3$) packets which will create a new capacity of

{$\frac{B*T}{S}+3$ - p * ($\frac{B*T}{S}+3$)} packets in the network. This capacity will be utilized by the N connections before the next proactive reduction happens. Every RTT each connection increases their congestion window by 1 packet as per (1) so N connections will increase the load by N packets every RTT and this gives the value of Proactive Reduction Epoch.

$$PR_E = \left\lceil \frac{(1-p)*(\frac{B*T}{S}+3)}{N} \right\rceil \qquad (19)$$

PR_E signifies the number of RTTs needed to use the available capacity created by a proactive reduction. It should be the ceiling of $\frac{(1-p)*(\frac{B*T}{S}+3)}{N}$ as the number of RTTs needed can't be fractional.

To calculate the average RTT value we have to consider that after the proactive reduction from $\frac{B*T}{S}+3$ to p * ($\frac{B*T}{S}+3$), the RTTs will not increase as long as the overall load is less than (B* T / S) as per (6a).

During this period the RTT will remain stable as per (4) to T. Whenever the load exceeds the capacity which the channel can sustain the queues will start growing and an increase in RTT will be perceived. Now we needed to calculate Proactive reduction epoch in terms of seconds rather than the number of RTTs. This is needed because the value of the RTT does not remain same throughout the proactive reduction epoch.

It can be visualized than inside a Proactive Reduction Epoch initially the RTTs will remain stable and then once the (B* T / S) threshold is crossed the RTT will start to dialate. So PR_{E_Total} can be divided into a period (i) period without RTT dialation and (ii) period with RTT dialation. Here PR_{E_Total}, $PRE_{RTTstable}$, $PRE_{RTT\_Dialation}$ are all expressed in terms of seconds.

$PR_{E\_Total} = PRE_{RTTstable} + PRE_{RTT\_Dialation}$

$$PRE_{RTTstable} = \left\lceil \frac{1}{N}\left\{\frac{B*T}{S} - p*\left(\frac{B*T}{S}+3\right)\right\} \right\rceil * T$$

$$= \left\lceil \frac{1}{N}\left\{(1-p)\frac{B*T}{S} - 3p\right\} \right\rceil * T \qquad (20)$$

$$PRE_{RTT\_Dialation} = 3*\left(T + \frac{Q_{avg}}{B}\right) \qquad (21)$$

$RTT_{avg} = (PRE_{RTTstable} + PRE_{RTT\_Dialation}) / PR_E$

$$RTT_{avg} = \frac{\left\lceil \frac{1}{N}\left\{(1-p)\frac{B*T}{S} - 3p\right\} \right\rceil * T + 3*\left(T+\frac{Q_{avg}}{B}\right)}{\left\lceil \frac{(1-p)*(\frac{B*T}{S}+3)}{N} \right\rceil} \qquad (22)$$

From (17) and (22), the overall throughput achieved by all the N connections expressed as packets/sec is given by

$$Throughput_{overall} = \frac{Cwnd_{average}}{RTT_{avg}}$$

$$Throughput_{overall} = \frac{\frac{(1+p)*(B*T+3S)}{2S}}{\frac{\left\lceil \frac{1}{N}\left\{(1-p)\frac{B*T}{S} - 3p\right\} \right\rceil * T + 3*\left(T+\frac{Q_{avg}}{B}\right)}{\left\lceil \frac{(1-p)*(\frac{B*T}{S}+3)}{N} \right\rceil}} \qquad (23)$$

The Individual throughputs will be $Throughput_i = Throughput_{overall} / N$

The RTTavg calculation in (22) has ceiling values both in the numerator and denominator as $\frac{1}{N}$ {(1-p)$\frac{B*T}{S}-3p$ } and $\frac{(1-p)*(\frac{B*T}{S}+3)}{N}$ may not be always an integral value.

To make it an integer let ε which is a small value such that 0 < ε < 1 which is equal to

$$\varepsilon = \left\lceil \frac{1}{N}\left\{(1-p)\frac{B*T}{S} - 3p\right\} \right\rceil - \frac{1}{N}\left\{(1-p)\frac{B*T}{S} - 3p\right\} \text{ or}$$

$$\varepsilon = \left\lceil \frac{(1-p)*(\frac{B*T}{S}+3)}{N} \right\rceil - \frac{(1-p)*(\frac{B*T}{S}+3)}{N} \qquad (24)$$

Considering ε the RTT_{avg} is given by

RTT_{avg} =

$$\frac{[\frac{1}{N}\ \{(1\text{-}p)\frac{B*T}{S} - 3p\ \} + \varepsilon\ ]\ *\ T\ +\ 3*(T+\frac{Q_{avg}}{B})}{\frac{(1-p)*(\frac{B*T}{S}+3)}{N} + \varepsilon} \quad (25)$$

The overall throughput given by $Throughput_{overall}$ =

$$\frac{(1+p)(BT+3S)*B*\{(1-p)(BT+3S)+\varepsilon NS\}}{2S\{BT(BT(1-p)-3pNS+\varepsilon NS)+3NS(BT+3S)\}} \quad (26)$$

The average value of the queue length Q_{avg}, when the proactive reduction is called for can be any value between $3 < Q_{avg} < Q$, where Q denoted the maximum size of the queue in terms of packets. But the minimum value of the queue to have three successive increase in RTT values is $Q_{avg} = 3$. So the value of Q_{avg} has been considered to be 3 in (25) to arrive at (26).

Equation (26) gives the overall throughput achieved by all the connections in the network in terms of the total bandwidth of the connection B, the ideal round trip time T, the size of the packets S and in terms of the number of concurrent flows N sharing the bandwidth. Individual flow throughput is given by

$Throughput_{i}$ =

$$\frac{1}{N} * \frac{(1+p)(BT+3S)*B*\{(1-p)(BT+3S)+\varepsilon NS\}}{2S\{BT(BT(1-p)-3pNS+\varepsilon NS)+3NS(BT+3S)\}} \quad (27)$$

4. SIMULATION BASED VALIDATION

We evaluate the performance of the proposed protocol in terms of goodput through simulations when several connections share the same link. We simulate the system as shown in the Fig.5, where N senders transmit data to N receivers through a satellite channel. The N streams are multiplexed in Earth Station A, whose buffer can accommodate K segments. In this experiment all the N senders are each connected to the Earth station A with a link of bandwidth 500kbps. All the N receivers are connected to Earth station B with a 500kbps link. We have taken N = 10, K = 25 segments, receiver window rwnd = 64 segments, the link between Earth Station A to B via satellite to be 5Mb and the RTT between the two stations as 550ms. All the results in this section have been obtained by considering the system behavior for T_Simulation = 550s which is 1000 times the round trip time value.

In this model we have not considered the packet errors in the channel as we want to show the efficiency of the proactive approach in handling congestion in the channel. The proactive approach basically solves the problems related to congestion so the error in the channels is not considered as in that case it will not be clear whether the packet loss is because of congestion or error in the channel. The channel errors will certainly degrade the throughput achieved because of the necessary retransmission of packets. The effect of channel errors on Proactive TCP has been extensively analyzed in [1] considering different packet error rates. It has been show in [1] that even in presence of channel errors Proactive TCP performs better than its peers like SACK [7] and Vegas [6].

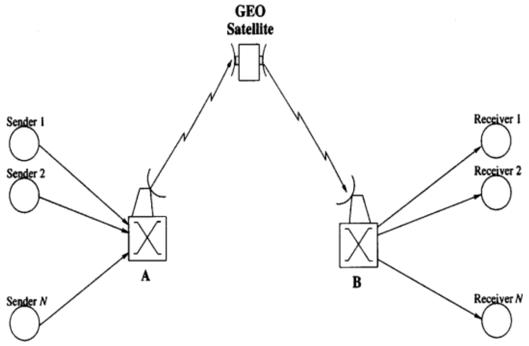

Fig.5 Simulation Scenario

4.1 Congestion window Evolution

Fig.6 shows the case where a single connection is sharing a 500kbps satellite channel bandwidth i.e. N =1. Here the BT/S = 35 or the bandwidth delay product of the link concerned is 35 packets. It can be seen that the congestion window is made to hover around 35 packets may be 36, 37 ,38 which coincides with assumption in the model that (BT/S+ 3) packets of overall load is the level where the proactive congestion reduction happen. This can be clearly seen here. Another interesting part of the protocol is that the congestion window throughout the simulation process does not fall appreciably like the reactive protocols where the congestion window is reduced to half when congestion is encountered. This shows how the protocol keeps the network load always within its tolerable limit. The Model well approximates the actual congestion window evolution process simulated in ns2[13].

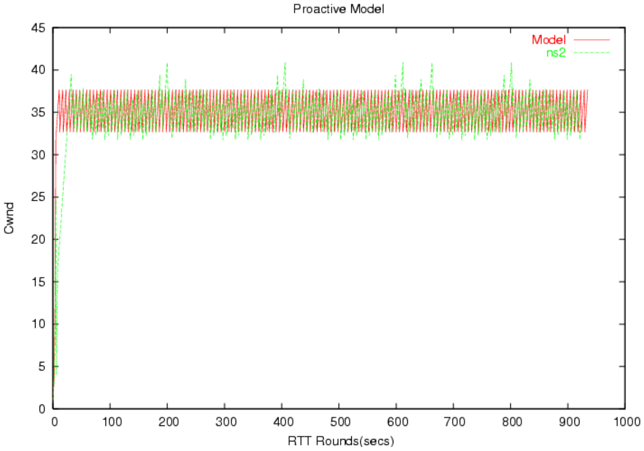

Fig6. Congestion Window evolution for ns2 simulation and model

4.2 Throughput validation for Single flow

The bottleneck link capacity as shown in the Fig5 is varied from 500 kbps to 5 Mbps and only one connection is made to occupy the entire channel. The graph in Fig.7 shows the achieved throughput in ns2 and as predicted by the mathematical model and also in Table1. The values are found to match closely.

Table1. Throughput Comparison with Single Flow

Bandwidth (Mbps)	0.5	1.0	2.0	3.0	4.0	5.0
NS2(kbps)	477.84	951.46	1881.48	2913.05	3842.40	4906.72
Model(kbps)	482.01	954.71	1885.59	2812.67	3738.72	4664.36

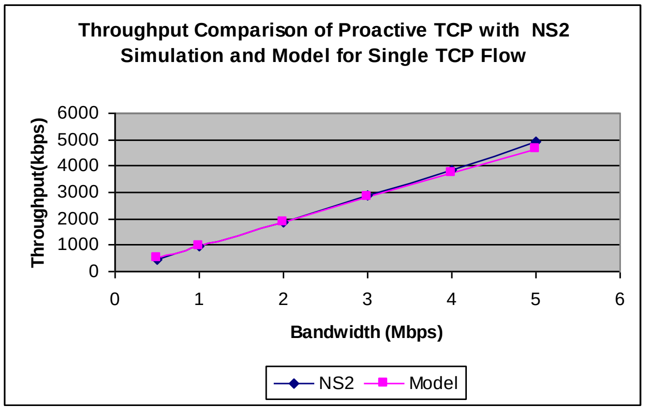

Fig. 7 Throughput comparison with ns2 and model for single flow

4.3 Throughput validation for Multiple Flows

The bottleneck link capacity as shown in the Fig. 5 is varied from 5 Mbps to 1Mbps and 10 connections are made to occupy the resources. The individual throughput achieved for all the connections have been averaged. The throughput predicted from the model using 10 connections are compared in Table2. From the graph it can be seen how closely the model approximates the simulated throughput in Fig8.

Table2. Throughput Comparison with Multiple Flow

Baandwidth (Mbps)	5.0	4.0	3.0	2.0	1.0
NS2 (kbps)	487.24	379.32	283.78	184.26	90.39
Model(kbps)	466.30	373.74	281.14	188.46	95.90

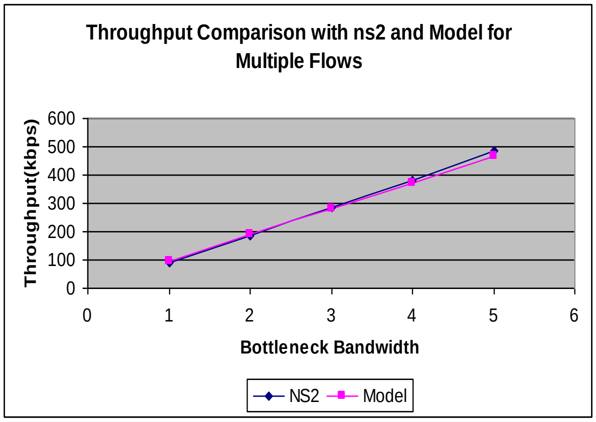

Fig. 8 Throughput comparison with ns2 and model for multiple flow

5. OPTIMAL VALUE OF P

To find the optimal value of p which maximizes the Throughput we have differentiated (26) with respect to p, equated to zero to arrive at a quadratic equation for p of the form $ap^2 + bp + c$, where the solution is given by

$$p = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \qquad (28)$$

The value of the coefficients a, b and c are given as follows

$$a = B(BT+3S)(BT+3NS) \qquad (29)$$

b =

$$-(BT+3S)\{2B^2T + 2B\varepsilon NS + 6NS * \frac{(BT+3S)}{T}\} \qquad (30)$$

c =

$$2\varepsilon NS(BT+3NS) + (\varepsilon NS)^2 + \frac{\varepsilon(3NS)^2}{BT} + B(BT+3NS)(BT+3S) \qquad (31)$$

This gives the optimal value in terms of B, T, S and N. Optimal value of p obtained using (28), (29), (30), (31) and different values of bandwidth, B and flows, N are given below in Table3, with T = 570ms and S=1040 bytes

Table 3. Optimal value of p for varying B and N

Bandwidth (B)(Mbps)	Number of Flow = 1	Number of Flow = 10
1	0.899093	0.764099
2	0.884147	0.836913
3	0.917961	0.873468
4	0.943010	0.897688
5	0.967043	0.916122

5.1 Simulation using the optimal penalty Factor

Considering the same simulation scenario of Fig5, different values of optimal p as shown in Table3 has been used to run single flow occupying different bandwidth and the results are shown in Table4 and graphically in Fig.9. A comparison is made with the throughput obtained using a penalty factor obtained empirically and that using the optimal values derived from this model.

Table 4. Comparison with optimal values for Single Flow

Bandwidth (Mbps)	0..5	1.0	2.0	3.0	4.0	5.0
P = 0.85 (kbps)	477.84	951.46	1881.48	2913.05	3842.40	4906.72
O.P(kbps)	489.82	992.89	1970.67	2976.72	3918.85	4952.50

The optimal penalty factor for 10 connections as shown in Table 3 has been used in the simulation and results compared with empirically derived penalty factors. The throughputs obtained by all 10 connections are averaged to generate the results in Table5 and Fig10.

Table 5. Comparison with optimal values for Multiple Flow

Bandwidth (Mbps)	5.0	4.0	3.0	2.0	1.0
P = .85(kbps)	487.24	379.32	283.78	184.26	90.39
O.P (kbps)	497.35	394.83	296.67	197.16	99.71

It can be seen that in all the cases the throughput obtained when using optimal values is better than that of the empirical values used.

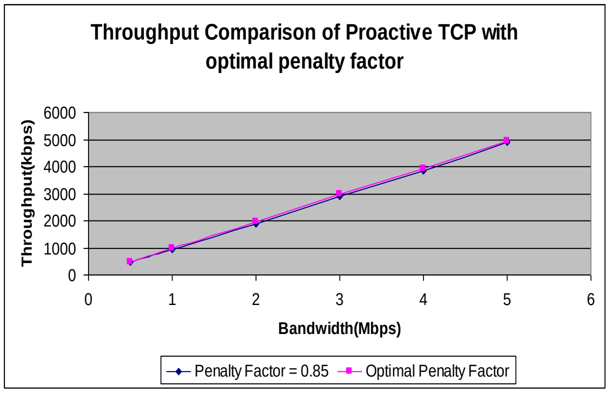

Fig. 9 Throughput using optimal penalty factor for single flow

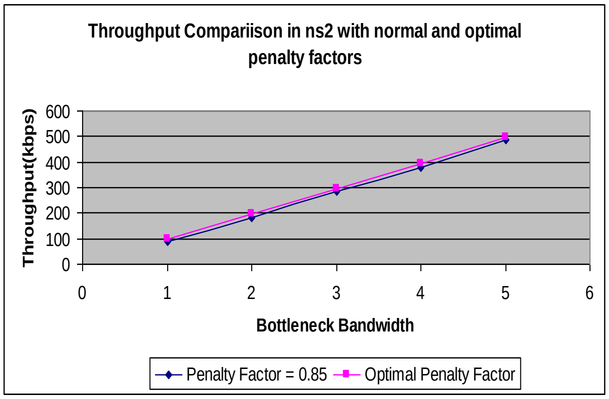

Fig. 10 Throughput using optimal penalty factor for multiple flow

6. CONCLUSION

In this paper it has been proved mathematically how a proactive approach to congestion control can provide a better utilization of network resources. The impact of assigning importance to the social self rather than selfish self has been critically examined in the paper through extensive simulation and analytical results. The objective of the paper is to show how with RTT initiated proactive reductions the overestimation of network capacity can be avoided. This has clearly been depicted in the simulation for congestion window evolution phase where the congestion window always hovers around the maximum network capacity and never falls to lower levels. This shows how the Proactive Protocol automatically controls flows in the network and always keeps the network utilization within its optimal operating point. In this paper the throughput achieved by the Proactive TCP has been modeled and tested with ns2 simulation and results are found to match closely. The implications regarding the choice of the penalty factor is also discussed and a mathematical function for the optimal value of the penalty factor which maximizes the throughput is evolved which is dependent on various network parameters like bandwidth, delay, packet size and number of concurrent flows. The overall objective of the paper as discussed in the earlier sections is to present a sound mathematical framework for Proactive Transport Protocols and establish the enhanced performance of Proactive Transport Protocols over conventional Reactive Transport Protocols.

7. ACKNOWLEDGEMENTS

Our special thanks to Mr. V.S.Palsule, D.D, SNAA, SAC, ISRO, Mr. A.P.Shukla, Head, ACTD, and Mr. N.G.Vasantha Kumar, Senior Scientist, ACTD, for providing constant encouragement towards the realization of this work.

8. REFERENCES

[1] Mohanchur Sarkar, K.K.Shukla, K.S.Dasgupta. 2010. A Proactive Transport Protocol for Performance Enhancement of Satellite based Network. (IJCA) International Journal of Computer Applications, Vol 1, Number16, pp. 114-121

[2] Jiang Wu, Hosam El-Ocla. 2004. TCP Congestion Avoidance Model with Congestive Loss. Proceedings 12th IEEE International Conference.Vol 1, pp. 3-8

[3] Ian F. Akyildiz, Giacomo Morabito, Sergio Palazzo. 2001. TCP-Peach: A New Congestion Control Scheme for Satellite IP Networks. IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 9, NO. 3, JUNE

[4] Ian F. Akyildiz, Xin Zhang, Jian Fang. 2002. TCP-Peach+: Enhancement of TCP-Peach for Satellite IP Networks. IEEE COMMUNICATIONS LETTERS, VOL. 6, NO. 7, JULY

[5] Ozgur B. Akan, Jian Fang, Ian F. Akyildiz. 2004. TP-Planet: A Reliable Transport Protocol for InterPlaNetary Internet. IEEE/SAC, vol 22, no 2, Feb. 2004, pp 348-61

[6] L. S. Brakmo, S. O Malley, L. L. Peterson. 1994. TCP Vegas: New Techniques for Congestion Detection and Avoidance. Proc. ACM SIGCOMM, pp. 24-35, October

[7] Mathis, M., J. Mahdavi, S.Floyd, and A.Romanow. 1996. TCP Selective Acknowledgment Options. RFC 2018, April

[9] J. Padhye, V. Firoio, D. Towsley, J. Kurose. 2000. Modeling TCP Reno Performance: A Simple Model and Its Empirical Validation. IEEE/ACM Trans. Networking, Vol. 8, No. 2, pp. 133-145, April

[10] M. Mathis, J. Mahdavi. 1996. Forward Acknowledgment: Refining TCP Congestion Control. Proc. ACM SIGCOMM 1996, pp. 281-292, Aug.

[11] S.C. Tsao, Y.C. Lai, and Y.D. Lin. 2007. Taxonomy and Evaluation of TCP-Friendly Congestion-Control Schemes on Fairness, Aggressiveness, and Responsiveness. IEEE Network, November

[12] R. Jain, D. Chiu, and W. Hawed. 1984. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. DEC, Res. Rep.TR-301

[13] UCB/LBNL/VINT Network Simulator [Online]. Available: http://www.isi.edu/nsnam/ns/