World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:3, No:4, 2009

Network State Classification based on the Statistical properties of RTT for an Adaptive Multi-State Proactive Transport Protocol for Satellite based Networks

Mohanchur Sakar, K.K.Shukla, and K.S.Dasgupta

Open Science Index, Computer and Information Engineering Vol:3, No:4, 2009 publications.waset.org/916/pdf

***Abstract*—**This paper attempts to establish the fact that Multi State Network Classification is essential for performance enhancement of Transport protocols over Satellite based Networks. A model to classify Multi State network condition taking into consideration both congestion and channel error is evolved. In order to arrive at such a model an analysis of the impact of congestion and channel error on RTT values has been carried out using ns2. The analysis results are also reported in the paper. The inference drawn from this analysis is used to develop a novel statistical RTT based model for multi state network classification.

An Adaptive Multi State Proactive Transport Protocol consisting of Proactive Slow Start, State based Error Recovery, Timeout Action and Proactive Reduction is proposed which uses the multi state network state classification model. This paper also confirms through detail simulation and analysis that a prior knowledge about the overall characteristics of the network helps in enhancing the performance of the protocol over satellite channel which is significantly affected due to channel noise and congestion.

The necessary augmentation of ns2 simulator is done for simulating the multi state network classification logic. This simulation has been used in detail evaluation of the protocol under varied levels of congestion and channel noise. The performance enhancement of this protocol with reference to established protocols namely TCP SACK and Vegas has been discussed. The results as discussed in this paper clearly reveal that the proposed protocol always outperforms its peers and show a significant improvement in very high error conditions as envisaged in the design of the protocol.

***Keywords*—**GEO, ns2, Proactive TCP, SACK, Vegas

I. Introduction

TCP has become the de-facto protocol standard for congestion control in the existing terrestrial Internet. However, experimental and analytical studies [18] confirm that the current TCP protocol variants have performance problems in long fat networks coupled with very high wireless channel errors like satellite networks [21], [20], [22]. Satellite based Networks are dominated by random packet errors which are not common in the wired counterpart. TCP protocols react to the lack of arrival of acknowledgments or duplicate ACK as a sign of congestion. So the congestion window is reduced which leads to unnecessary throughput degradation. It is a challenge for the network researchers and protocol developers to find means to differentiate the cause of the DUP ACK arrival. Generally probing is done in protocols like Peach [1], Peach+ [2], TP-Planet [3] and RCS [4] but at the cost of bandwidth used for the low priority dummy packets.

Majority of the Transport Protocols tries to ascertain the condition of the network in terms of an estimation of the prevailing congestion in the network [10]. This is because of the fact that the basic paradigm on which the conventional TCP protocols evolved was based on a wired connectivity where congestion was the primary concern. The channel noise was not taken into account mainly due to the fact that wired connectivity can be through of as an errorless channel. So the network states were broadly classified into two states congested and uncongested. But when TCP Protocols are used for Satellite based Networks the effect of wireless channel errors also become significant [21]. Hence this assumption of two state concept does not lead to the most optimal performance. It is obvious from the above mentioned reasons that use of TCP over Satellite channel need to address equally the effect of congestion and channel noise which are important detrimental factors for performance degradation. To handle the problem squarely one need to consider Multi State representation of network condition. There is a need to develop a model to handle and represent Multi State network conditions for satellite based TCP/IP networks. In order to formulate the model and understand the various system parameters which control it, the model need to be explicitly evolved and analyzed.

The inferences from these analyses as discussed in this paper are used in an innovative way to work out an Adaptive Multi-State Proactive Transport Protocol with a view to improve the overall network performance over a Satellite channel which has high degree of channel noise and appreciable congestion. In order to formulate the Multi-State Network classification model detail simulation based

Mohanchur Sarkar is with the Space Application Centre, ISRO, Ahmedabad 380015, INDIA, (Phone No: +91-079-26912489; fax: +91-079-26915818, (e-mail: mohanchur@yahoo.com, msarkar@sac.isro.gov.in,)

K. K. Shukla is with the Computer Engineering Department, University of BHU, Varanasi, 221005 INDIA, (e-mail: kkshukla.cse@itbhu.ac.in)

K.S.Dasgupta is with the Space Application Centre, ISRO, Ahmedabad 380015, INDIA, (e-mail: ksd@ sac.isro.gov.in).

Open Science Index, Computer and Information Engineering Vol:3, No:4, 2009 publications.waset.org/916/pdf

experiments in ns2 [16] have been carried out to ascertain the combined effect of congestion and channel error on the RTT values pertaining to a connection. The inference from these simulation experiments has been used in the evolution of a Statistical model for network state classification based on RTT values.

The paper is organized as follows. We introduce the RTT Analysis for Multi State Model in Section II. In Section III, we propose the statistical formulation of model parameters. Section IV describes the hierarchical network state classification method. Section V describes the Adaptive Multi State Proactive Transport protocol. Section VI describes the design considerations of the proposed protocol. Section VII details the Test and Evaluation methodology. Finally, in Section VIII we conclude the paper.

II. RTT ANALYSIS FOR MULTI STATE MODEL

The primary goal of the model is to carry out multi state classification of network states with a view to differentiate (i) No Congestion No Error, (ii) No Congestion High Error, (iii) High Congestion No Error and (iv) High Congestion High Error conditions of the network. This will be a significant departure from the conventional two state network classification model with states (i) No Congestion and (ii) Congestion. Using the four states model the combined effect of congestion and corruption on the network can be ascertained. This model will be primarily worked out to handle the performance degradation of TCP over satellite channel where there is coexistence of channel noise and congestion. For the formulation of the model simulation based experiments have been carried out in ns2 [16] to study and analyze the impact of congestion and channel errors on the model parameters. The inference of these analysis will help us in understanding a set of network system parameters which can be properly incorporated in the proposed Adaptive Multi State Proactive Transport Protocol.

A. RTT based Modeling Philosophy

When a TCP connection is in progress, the sender has no idea about what is happening in the network. It only receives the ACKS, sometimes duplicate ACKS and encounters timeout when the ACKs do not come within an anticipated time. From the received ACKs, the RTT is the only information available to the sender which can be used for a proactive decision making regarding the state of the network.

The main philosophy of our approach is that congestion and channel errors are two different phenomenons, one being caused by overestimation of network capacity and the other being the inherent characteristics of any wireless channel and depends on different communication related parameters. So the effect of congestion and channel errors on the pattern of RTT values must be different. In this paper, an attempt has been made to exploit this difference in predicting the network states, by using some statistical parameters of the available RTT values. When a TCP connection progresses, the congestion in a network does not grow all of a sudden. The queue in the routers start growing and that leads to more packet delay or an increase in the experienced RTT. Then after a certain point when the network cannot adjust to the increased flow of packets a congestive meltdown is perceived [29] in case of a Reactive TCP Protocol like Tahoe, Reno [7], New Reno, SACK[6], FACK [10], Peach[1], Peach+[2], TP-Planet[3] or a correction takes place in terms of transmission speed as in Vegas[5] and Proactive TCP[28]. So the connection moves through periods of no congestion state, through incipient congestion state to high congestion state conditions. The point which we are trying to capitalize on is, whatever event happens in the network like high congestion events or high errors keeps its imprint on some parameters of the network.

B. Use of RTT Mean

The RTT in general is considered to be an independent random variable [11]. While a TCP connection proceeds, if we measure the individual RTT values for every window for all the segments transmitted in that window, it will not convey much information regarding the condition of the network, as the transient RTT values are dependent on many dynamic network parameters which changes so frequently that no stable conclusion can be drawn from them. Moreover as suggested in [27], the frequency with which we sample the RTT values is generally less than the Nyquist's Sampling frequency requirement if we consider the RTT values as a time varying signal. So as the sampling is done at a rate lower than needed a proper reconstruction of the signal will not be possible and there is a chance to respond to values corrupted with noise. Another important point is that in a network there are many connections sharing a common bandwidth, so an increase or decrease in RTT cannot be attributed to be happening because of that particular connection and taking action on the individual RTT values may not generate optimal conclusion [27]. So we have considered the mean of all the RTT values in a window, which convey more information or may be thought of as a more representative value of the RTT for that specific window to be used in our RTT based model.

C. Simulation based Experimental Results

In this paper, we have performed some simulation based experiments using ns2[16] with the simulation setup of 10 senders transmitting to 10 receivers over a GEO link as described in Section VII, to find out the impact of network congestion and packet error rates on the RTT values. These simulation based experiments are needed in the formulation of the Multi State representation of network condition. Three classes of experiments have been performed, (i) considering an uncongested network, (ii) considering a congested network and (iii) progressive levels of network congestion but with very less channel errors.

D. Analysis of Results

In Fig.1 to Fig. 9 the mean RTT values are plotted with simulation time for a TCP connection as the simulation

Open Science Index, Computer and Information Engineering Vol:3, No:4, 2009 publications.waset.org/916/pdf

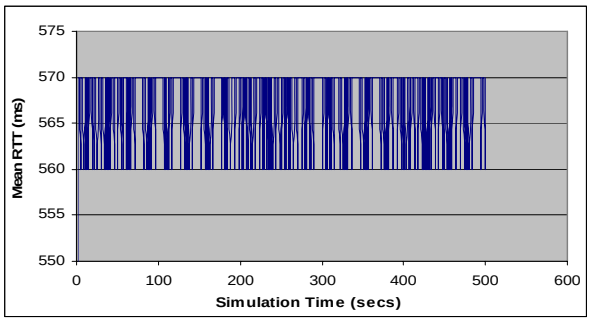


Fig. 1 PER = 0.00 Queue = 20 Max Receiver Window = 20

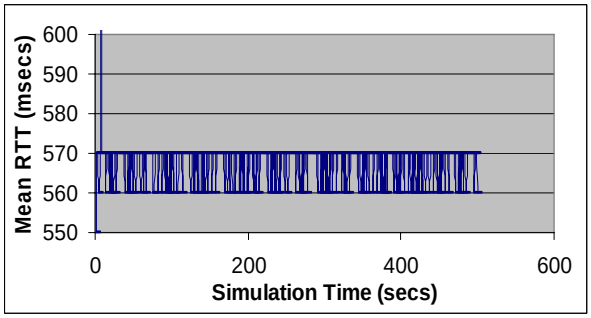


Fig. 2 PER = 0.001 Queue = 20 Max Receiver Window = 20

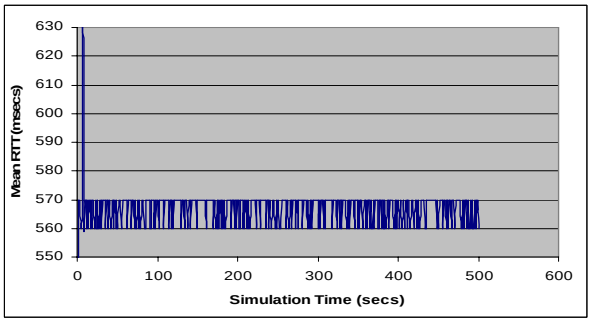


Fig. 3 PER = 0.01 Queue = 20 Max Receiver Window = 20

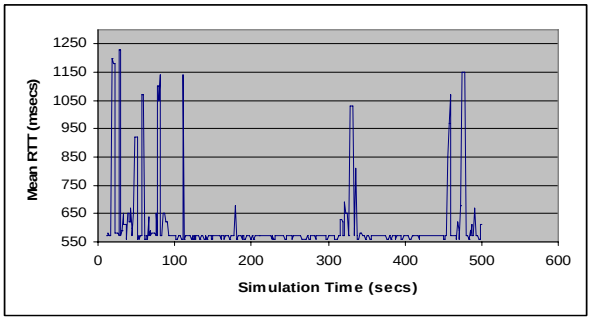


Fig. 4 PER = 0.1 Queue = 20 Max Receiver Window = 20

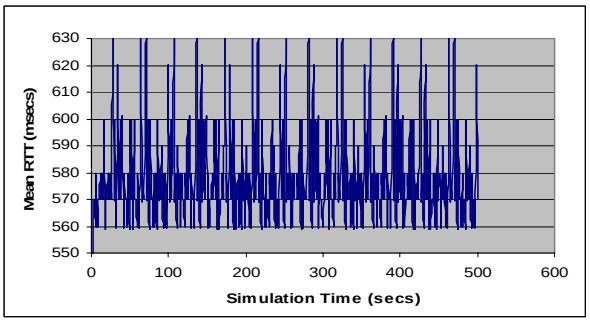


Fig. 5 PER = 0.00 Queue = 64 Max Receiver Window = 64

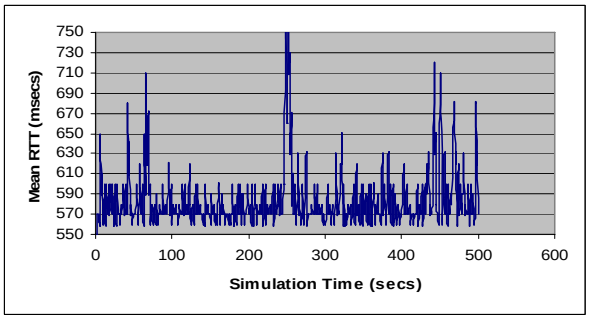


Fig. 6 PER = 0.001 Queue = 64 Max Receiver Window = 64

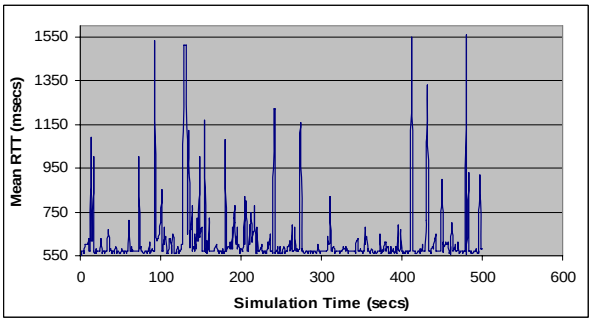


Fig. 7 PER = 0.01 Queue = 64 Max Receiver Window = 64

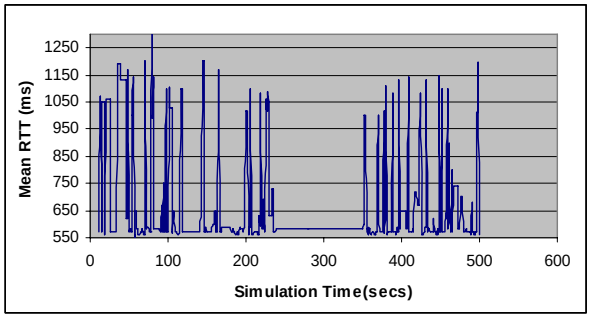


Fig. 8 PER = 0.1 Queue = 64 Max Receiver Window = 64

progresses. First, we have considered the packet error rate to be 0 and it can be seen from Fig. 1 that the mean RTT remains in a closed range of 560 to 570 ms. On the contrary in Fig. 5 even with PER 0, the mean RTT values are more dispersed which shows the effect of congestion on the RTT values. In both the network conditions we have gradually increased the packet error rate from PER 0.001 to PER 0.1 and it has been observed that as we increase the PER for uncongested network the values are seen to have more frequent flickers and dispersion increases. In a congested network condition the effect is more pronounced than its uncongested counterpart for the same PER value. It has also been observed that congested network condition leads to an increase in the DC value of the concerned parameter, larger dispersion and flickering of mean RTT.

In Fig. 9 to ascertain the effect of congestion on mean RTT values we have intentionally infused higher levels of congestion in the network, by chocking the available bandwidth to the satellite to lower values of 4Mbps, 3Mbps down to 1Mbps where the optimum bandwidth required is 5 Mbps as described in Section VII in detail. Fig.9 directly depicts the high correlation, the DC value of the RTT mean

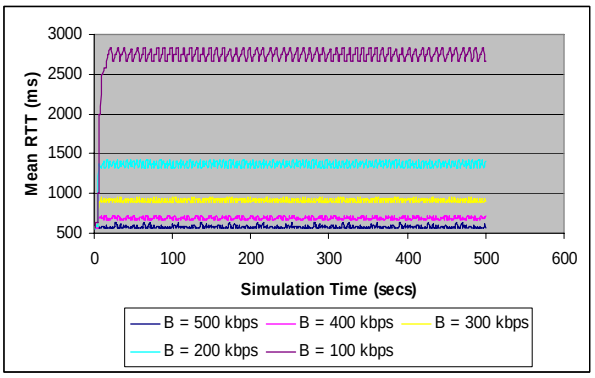


Fig. 9 RTT Mean Variation For Different Congestion Levels

has with increasing levels of congestion. The simulation experiment used a GEO satellite link having RTT around 550 to 570 ms. In the simulation scenario decreasing bottleneck bandwidth between the two earth stations the individual maximum bandwidth shared by the connections can be though of as reducing from 500 kbps to 100 kbps. The mean RTT values are seen to increase and to be hovering in the range of 570 ms, 700ms, 900ms, 1400ms and 2700ms corresponding to the decrease in link capacity and increasing degree of congestion.

The inferences which can be drawn from these experiments are given below as (i) From the first experiments it can be concluded that no congestion and no error leads to very less dispersion of the mean RTT values, dispersion increases with increasing PER (ii) From second experiment it can be concluded that under congested condition, the increase in packet error rate leads to more dispersion of mean RTT values. (iii) From the third experiment it can be concluded that if channel error is very less the degree of congestion is directly related to the increase in DC value of the RTT Mean.

So it can be concluded that RTT has multivariate correlation with congestion and PER prevailing in the network.

III. Statistical Formulation of Model Parametrs

This section describes the evolution of the model from the inferences discussed in Section II. As discussed in the previous section the congestion in the network and the packet error rate in the channel have a reflection in the pattern of the mean RTT values. In this section, considering this inference we have developed a model to classify the short term and long term characteristics of a network by estimating the presence and absence of congestion and corruption in the network. Considering the combinatorial combination of the presence of congestion and corruption as binary variable 0 and 1, the network can be considered to be in one of the four state 00, 01, 10, 11 where,

state 00 signify - No Congestion No Error
state 01 signify - No Congestion High Error
state 10 signify - High Congestion No Error
state 11 signify – High Congestion High Error

Depending on the dynamics of the data passing through the network, error encountered during the transfer process and the reaction of the congestion control process towards the undelivered packets the network moves through the above mentioned states. From the available RTT values we have intended to determine certain statistical parameters, which represent the extent of congestion & corruption in the network. These parameters when combined through a hierarchical decision making process helps in classifying the network into the above mentioned states. The classification model using the statistical parameters along with their definitions is presented below.

A. RTT based Network State Classification Model

The total duration of a TCP connection can be visualized to be divided into a series of windows, where at the start of each window the number of TCP segments to be transmitted in that window is governed by the value of the prevailing congestion window (cwnd). TCP sends all the segments up to cwnd and waits for acknowledgment of the segments to return to determine the value cwnd for the next window. So for a given window, there will be a series of RTT values obtained corresponding to all packets transmitted in that window.

Lets us assume that there are m number of windows in the duration of a TCP connection denoted by W_1, W_2… W_i … Wm. Let us define the following variables

W_i denote the ith window, $0 < i < m$

$cwnd_i$ denote the value of cwnd for the ith window,

$RTT_{i,j}$ is the RTT measurement for the jth packet for the ith window, $0 < i < m$, $0 < j < cwnd_i$

So we have a series of RTTi,j values corresponding to each window Wi, which are the independent values from where we try to define the following variables

$$MW_i = \frac{\sum_{j=0}^{cwnd_i} RTT_{i,j}}{cwnd_i} \quad (1)$$

MWi denote the mean of the RTT values for the ith window Wi

$$M_k = \frac{\sum_{j=i-k}^{i} MW_j}{k} \quad (2)$$

M_k denotes the window average of the mean values MW_j for the last k windows.

Open Science Index, Computer and Information Engineering Vol:3, No:4, 2009 publications.waset.org/916/pdf

Open Science Index, Computer and Information Engineering Vol:3, No:4, 2009 publications.waset.org/916/pdf

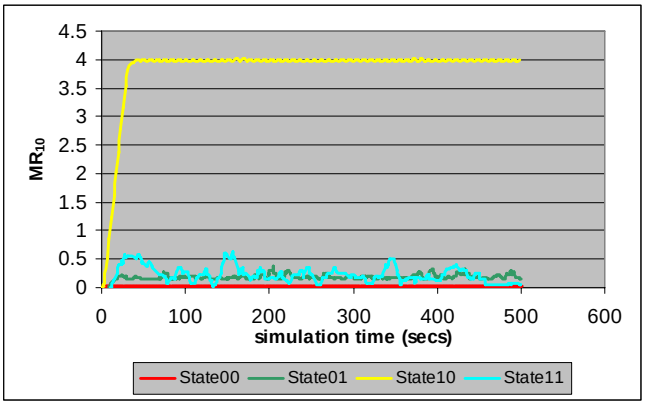


Fig. 10 Mean Rise Window Average Length = 10

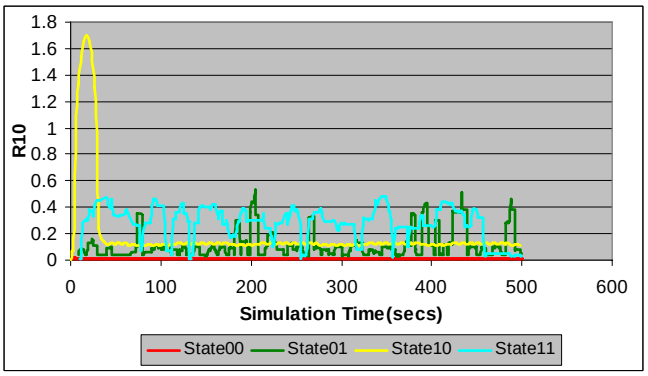


Fig. 11 Ratio Window Average Length = 10

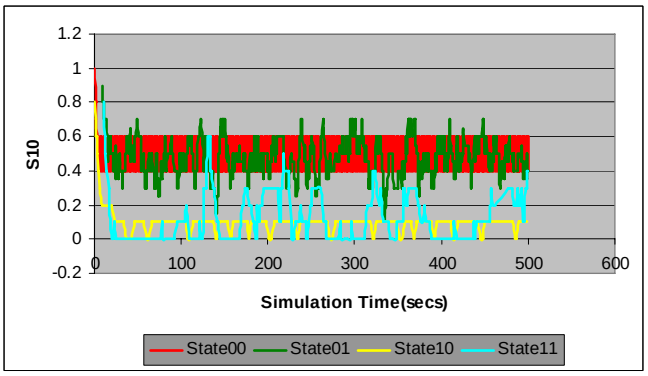


Fig. 12 Stability Window Average Length = 10

$$MR_k = \frac{M_k - IRTT}{IRTT} \tag{3}$$

MR_k denote the normalized rise of the mean RTT values from the ideal RTT values. IRTT can be determined from previous knowledge of the end to end delay and is given by the minimum of all RTT values measured for that specific connection. So IRTT in the beginning will be equal to the RTT values obtained through the SYNC message exchanged during connection setup and gradually can be refined with new values. MR_k is an estimation of the short term condition of the network load.

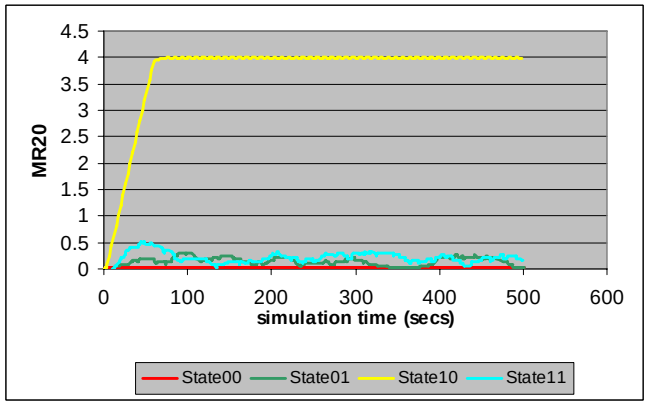


Fig. 13 Mean Rise Window Average Length = 20

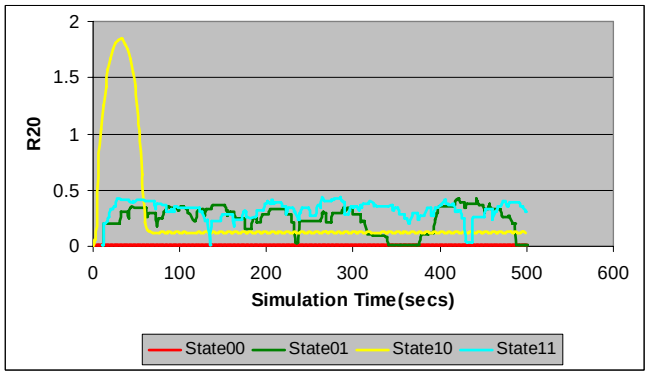


Fig. 14 Ratio Window Average Length = 20

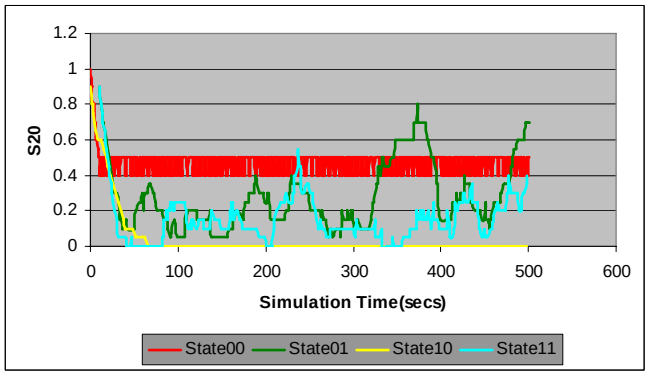


Fig. 15 Stability Window Average Length = 20

$$M_i = \frac{\sum_{j=0}^{i} MW_j}{i} \tag{4}$$

In (2) if we replace k with i then we get the long term mean M_i which is equal to the mean of all the mean values MW_j obtained for each window from the beginning. So the M_k values give the short term characteristics of the network and M_i gives the long term characteristics as it is the integration of all the values the connection has experienced from its inception. Similarly MR_i denotes the long term mean rise as given below where i denote the prevailing window. So as the connection progresses, at each instant if the prevailing window is i, then MR_i denotes the rise of the mean from the

Open Science Index, Computer and Information Engineering Vol:3, No:4, 2009 publications.waset.org/916/pdf

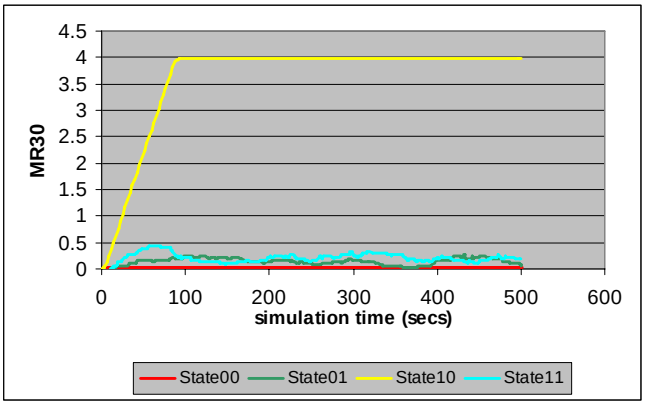


Fig. 16 Mean Rise Window Average Length = 30

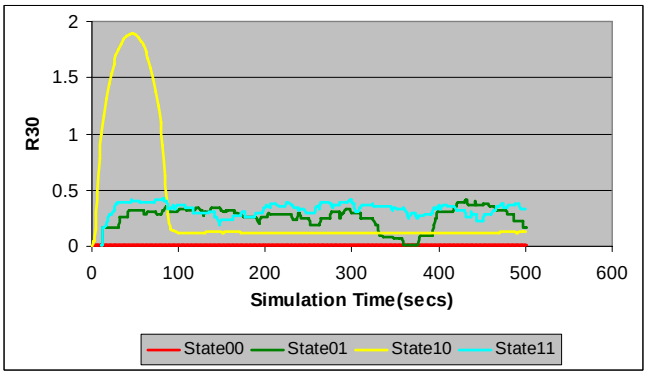


Fig. 17 Ratio Window Average Length = 30

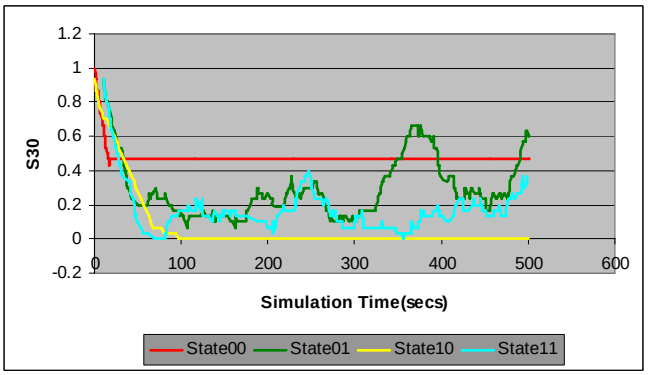


Fig. 18 Stability Window Average Length = 30

ideal value considering values from beginning of connection.

$$MR_i = \frac{Mi - IRTT}{IRTT} \tag{5}$$

Next we calculate the standard deviation of the mean rtt values, MW_j for the last k windows given by

$$\sigma_k = \sqrt{\frac{\sum_{j=i-k}^{i}(MW_j - M_k)^2}{k}} \tag{6}$$

Here k denotes the number of last window values considered.

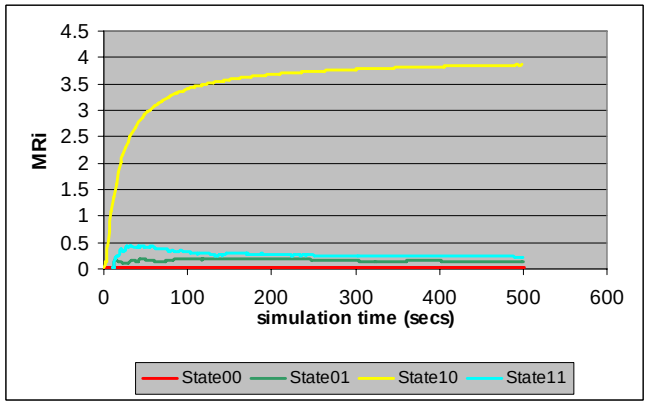


Fig. 19 Mean Rise Window Average Length = i

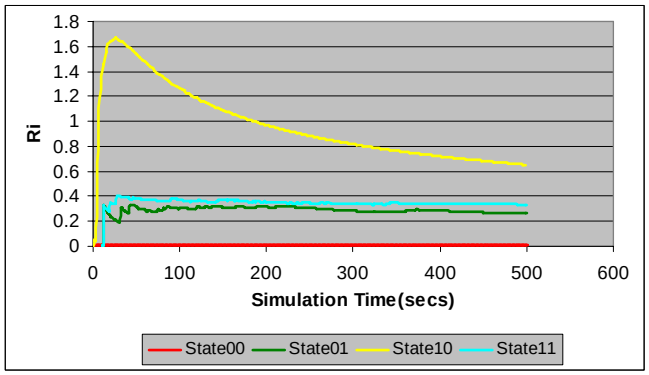


Fig. 20 Ratio Window Average Length = i

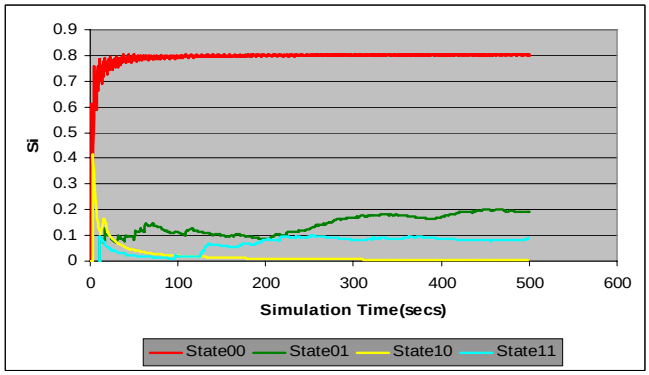


Fig. 21 Stability Window Average Length = i

$$\sigma_i = \sqrt{\frac{\sum_{j=0}^{i}(MW_j - M_i)^2}{i}} \tag{7}$$

When k is replaced by i, which is the prevailing window index this gives the long term standard deviation.

$$R_k = \frac{\sigma_k}{IRTT} \tag{8}$$

R_k denotes the ratio of the standard deviation to the ideal RTT which signifies the extent to which the RTT values have been scattered from the ideal condition. Ri denotes the long term ratio.

$$R_i = \frac{\sigma_i}{IRTT} \quad (9)$$

The parameter S_k denotes the ratio in which the intermediate Mean RTT values MW_j and MW_{j+1} have not changed beyond a threshold α to the total number of windows considered. In our case α is taken as 1% of previous RTT Mean so, α = 0.01 * MWj. S_k denotes the short term stability of the network.

```
Stability (k) {
for (j = i-k; j = i; j++) {
  if ((|MW_{j+1} – MW_j|) < α)
        cnt++;
    α = 0.01 * MWj;}
    s = cnt / k;
 return s;
}
```

When k is replaced by i, S_i denotes the stability of the connection from the beginning.

In the RTT based Model for network state classification we have defined some statistical parameters from the measured rtt values which convey different meaning about the short term and long term characteristics of the network. So corresponding to each window Wi we can think of the set of values $\{MR_{10}, MR_{20}, MR_{30}, MR_i\}$ for Mean Rise, the set $\{R_{10}, R_{20}, R_{30}, R_i\}$ for Ratio and for stability $\{S_{10}, S_{20}, S_{30}, S_i\}$ to signify the short term and long term characteristics of the network. We have shown through simulations how these parameters vary with time.

B. Short Term Network Characteristics

The short term characteristics of the network can be obtained by considering a window average of some fixed number of past windows. This will give us a more recent condition of the network. We have simulated considering the last 10, 20 and 30 windows which are displayed in the graph below. All the three parameters considered in the protocol like mean rise, ratio and stability is seen to vary as shown in Fig. 10 to Fig. 18.

C. Long Term Network Characteristics

From the Fig. 10 to Fig. 21 it can be seen that as the value of the parameter k increase from 10 to 20 to 30, the standard deviation of all the individual plotted statistical parameters is seen to reduce. This is obvious considering the principle of the sampling theory, that a larger sample reduces the standard deviation of the sample mean and leads to a more accurate estimation of the population concerned. So, the long term estimate MRi, Ri and Si is seen to have much less dispersion in Fig. 19, Fig. 20 and Fig. 21 as the simulation time progresses, thereby characterizing the network into its different states. So the variable MRi, Ri and Si can be considered as the network state variables and for each progressive window, get updated as per the equations described above.

IV. Hierarchial Network State Classification

This section describes the network state classification logic along with the classification of the statistical parameters described in Section III. The equations to derive the thresholds have been explained considering values derived from simulation experiments. Table I, Table II and Table III contain values of parameter MRi, Ri and Si respectively calculated from simulations, considering the scenario described in Section VII for a Geo Satellite Network. In Section V it will be shown how the Parameter Adaptation Algorithm automatically generates the Tables described in this section. A pictorial representation of the parameter classification logic along with multi state hierarchical network classification logic is also presented in this section.

A. Mean Rise Classification

Table I gives the minimum and maximum value obtained by the Mean Rise, MRi parameter through the simulation experiment.

TABLE1
MEAN RISE VALUES FOR GEO NETWORK (MRi)

State	00	01	10	11
Min	0.031	0.1	0.40	0.20
Max	0.10	0.2	3.96	0.50

Let us define the following thresholds LMRi and HMRi which differentiate the MRi value into low (L), medium (M) and high (H) ranges as shown in Fig. 22

$$LMR_i = Max\{MR_i State00, MR_i State01\} \quad (10)$$

$$HMR_i = Max\{MR_i State11\} \quad (11)$$

LMRi denotes the lower threshold is given by the max value obtained by MRi in 00 and 01 states. So when the value of MRi is below this threshold we can assume that the network is in 00 or 01 state. HMRi denotes the upper threshold is given by the max value obtained by MRi in state 11. When the value of MRi is between LMRi and HMRi the network is estimated to be either in state 10 or 11. MRi values higher than HMRi denotes the high congestion no error state of 10. Fig. 22 shows the classification of network states using MRi.

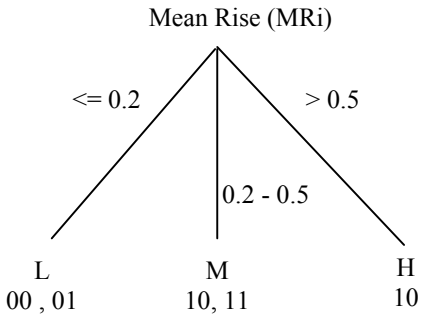


Fig. 22 Classification of Network states using Mean Rise MRi

Open Science Index, Computer and Information Engineering Vol:3, No:4, 2009 publications.waset.org/916/pdf

Open Science Index, Computer and Information Engineering Vol:3, No:4, 2009 publications.waset.org/916/pdf

B. Ratio based Classification

The following table gives the minimum and maximum value obtained by the Ratio parameter Ri through the simulation experiment. We define a threshold

$$LR_i = Max\{R_i State00\} \quad (12)$$

TABLE II
RATIO VALUES FOR GEO NETWORK (Ri)

	00	01	10	11
Min	0.013	0.23	0.179	0.35
Max	0.2	0.3	1.2	0.40

LRi is used as a threshold which divides the ratio values into two broad cases low (L) and high (H) and any value lower than the threshold can be considered to be a sign that the network is in state 00. If values higher than this are obtained then the network can be in either of the states 01, 10 and 11 as shown below.

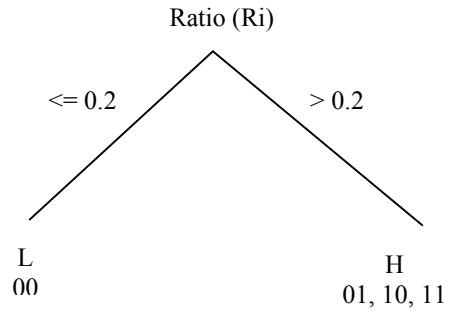


Fig. 23 Classification of Network states using Ratio Ri

C. Stability based Classification

The following table shows the values of the stability parameter obtained through simulation experiment. We can define some threshold, which differentiate the stability parameter into Low (L), Medium (M) and High (H) values classifying the network into states as shown in the Fig. 24. Let us define the thresholds

$$LS_i = Max\{S_i State10\} \quad (13)$$

$$HS_i = Min\{S_i State00, S_i State01\} \quad (14)$$

TABLE III
STABILITY VALUES FOR GEO NETWORK (Si)

	00	01	10	11
Min	0.40	0.16	0.00	0.02
Max	1.0	0.4	0.02	0.16

Si values lower than LSi will be considered low denoted by L which is typically seen in state 10 and value between LSi to HSi is typically seen in state 11 and considered to be medium denoted by M and values higher than HSi means a high stability factor denoted by H typically seen in state 00 and 01.

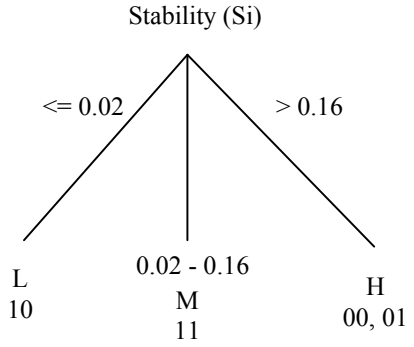


Fig. 24 Classification of Network states using Stability Si

B. Classification Combining all Parameters

For the classification process we need to determine the threshold values for all the parameters described MRi, Ri and Si. For a type of network like GEO satellite network the values of the parameters can be predetermined. For a more accurate threshold calculation there has to be an adaptive way of finding out the thresholds values. The protocol has to be run on the network concerned for a sufficiently long span of time in predetermined network conditions. The values of the different parameters for all these different network conditions can be stored till the parameters become stable or their standard deviation is almost zero. It has been seen in Fig. 19, Fig. 20 and Fig. 21 how the parameters lead to stable value conveying long term characteristics of the network. Once these stable values of these parameters are obtained the classification thresholds can be determined using the equations (10) to (14) discussed above.

Now we define the classification approach where in the beginning the network is considered to be in any one of the four states 00, 01, 10, 11, Then we check the MRi value and the estimated state is partioned to {00, 01}, {10, 11} or 10 state depending on whether the mean rise is very less, moderate or high. A very high mean rise is only seen in state 10 so in case a very high value is obtained it can be directly inferred that the state is 10. For other cases to be more certain a checking is done with the Ri parameter. A low and high value of Ri partitions the set {00, 01} to 00 and 01 respectively and set {10, 11} becomes unpredictable if Ri is low. To be more precise we check for the stability parameter Si to be high with state 00 and 01. For the set {10,11} the Si parameter classifies it to state 10 when low and state 11 when having moderate values. In cases the proper combination of the parameters are not found the network is said to be unclassified which is shown by X in Fig. 25. In this context one point is worth mentioning that if more number of unclassified states are arrived from the network classification algorithm that signifies that the threshold value calculated are not properly tuned so a recalculation is needed in that case.

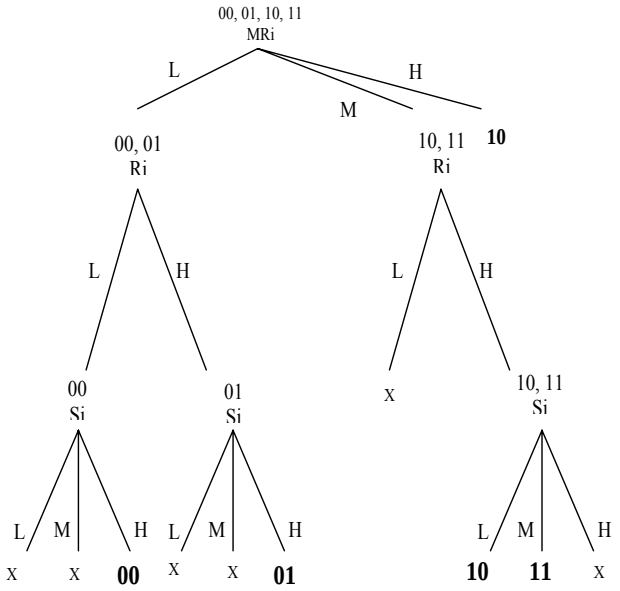


Fig. 25. Classification combining MRi, Ri and Si

V. Adaptive multi state proactive TCP protocol

A. Brief Working of Adaptive Multi-State Proactive TCP

The protocol uses a proactive approach and is composed of new approaches like Proactive Slow Start, Proactive Congestion Avoidance, State based Error Recovery, State based Timeout Action algorithms along with traditional TCP algorithms like Fast Retransmit. The philosophy of the protocol is that certain statistical parameters of the network can be slowly adapted for better proactive actions. With a prior knowledge about the characteristic of the network the performance of the protocol can be greatly enhanced specially during challenging environments of high wireless link errors. In Fig. 26 the overall mechanism of the proposed Adaptive Multi-State Proactive TCP scheme is depicted.

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 acknowledgments 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 [20]. 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 [28]. All the AIMD TCP protocol variants like Tahoe, Reno, NewReno, SACK [6], FACK [10], Peach [1], Peach+ [2], TP-Planet [3] fall in this category. Reactive algorithms tries to solve the problem but don't consider why the problem happens. 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. TCP Vegas [5] is a proactive protocol which is also recommended by the CCSDS SCPS-TP [17] for use in satellite based networks.

In this section we introduce an Adaptive Multi State Proactive Transport Protocol, which is an end-to-end solution to improve the throughput performance in satellite networks.

The Network State Classification Algorithm is the heart of this Adaptive State based protocol. For every window after the ACKs for all the transmitted packets are received before the start of a new round of packet transfer, Network State Classification is performed which estimates the state in which the network is presently or an unclassified state using the Network Classification Algorithm as described in Fig 31.

The protocol starts with the Proactive Slow Start phase where the slow start threshold is kept at half the receiver window, the congestion window to one and the state of the protocol is No Congestion No Error ie state 00. The receiver window signifies the maximum number of segments the receiver can accommodate. The congestion window is increased by one with each received ACK so that the congestion window doubles every RTT until the slow start threshold is reached. But the mean RTT is also checked during this phase and if RTT increase for three successive windows is detected the congestion window is reduced by an Adaptive penalty factor shown in Fig. 40. After slow start threshold is crossed the protocol moves to the Proactive Congestion Avoidance phase where it increases the congestion window by 1/cwnd for every reception of ACK as in traditional Congestion Avoidance Algorithm. Now if an acknowledgment is not received for a transmitted segment within its retransmission timeout period, the timer expires and the algorithm enters the State based Timeout Action phase. As shown in Fig. 37 here depending on the state of the network the timeout action is taken. If the state is any among 00, 10 or 11 then the congestion window is reduced to one and Proactive Slow Start is initiated. If the state is 01 then the congestion window is not reduced and the protocol remains in Proactive Congestion Avoidance, only the packets not acknowledged are retransmitted.

During the Proactive Congestion Avoidance the mean RTT is checked every time and if an increase in RTT for three successive congestion windows is detected the congestion window is reduced by the adaptive penalty factor using State based Proactive Action Algorithm as shown in Fig. 41. For every window the Network State Classification also predicts the state of the network.

Now if three duplicate ACKs are received the protocol calls the Fast Retransmit algorithm and a retransmission is done of the lost segment and the protocol moves to the State based Error Recovery Algorithm as shown in Fig. 39. In this phase if the protocol is in state 10 or 11 which signals that the network has some congestion and a packet is not received by the receiver then the chance of the packet being lost by congestion is more. So the congestion window is reduced to

Open Science Index, Computer and Information Engineering Vol:3, No:4, 2009 publications.waset.org/916/pdf

Open Science Index, Computer and Information Engineering Vol:3, No:4, 2009 publications.waset.org/916/pdf

half and protocol moves to the Proactive Congestion avoidance phase. If the state of the network is 00 then the congestion window is reduced to ¾ of this value. No reduction in congestion window is done when the state of the network is 01 which signifies that the loss is error initiated. An unclassified state is considered as state 11 because in state 11 all actions are conservative and this prevents from performance degradation during unsuccessful classification.

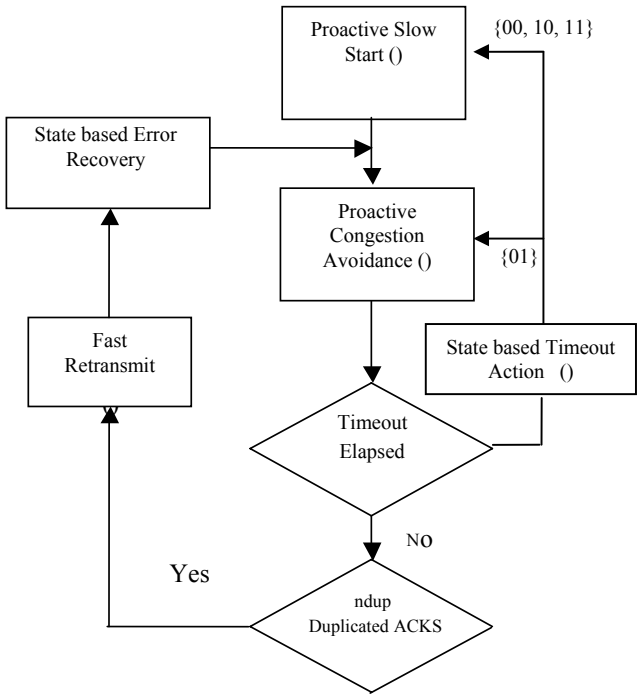


Fig. 26 Flow Start of Adaptive Multi State Proactive TCP

The Adaptive Multi State Proactive TCP Protocol uses some concepts of Proactive TCP [28] Protocol designed by the authors and is best suited to work in high error conditions.

B. Overall Operation of the Protocol

The overall operation of the protocol can be classified into a Training Phase and an Operational Phase. During the Training Phase the protocol learns about the characteristics of the network by adapting the statistical parameters and calculates the threshold parameters, which are eventually used for network state classification. The basic algorithm is presented below in Fig. 27.

Before the Adaptive Multi State Algorithm can be used it is necessary to have the threshold values for the network state classification process. For a particular type of network these values can be predetermined where the traffic dynamics are less. For example we have calculated the value of the thresholds for a GEO based network in our simulation based experiments with an idea about the ideal RTT value for a GEO satellite which is 550ms. With predetermined threshold values pertaining to the type of network in which the protocol is supposed to run, the training process may be skipped. But for networks, which have a more dynamic pattern of traffic, before the protocol can be used a training of the protocol is necessary to calculate the threshold values of the parameters. This is a familiarization of the network dynamics to the protocol under different conditions of congestion and error. The network has to be run through varying levels of congestion and corruption so as to create the different network states while the parameters are recorded. In an overview the main algorithms to be used are given below in Fig. 27. Basically using the Parameter Adaptation process the data collection will happen followed by Threshold_Calculation Algorithm to calculate thresholds. These two algorithms are part of the initial training process.

Once the thresholds are obtained, during the protocol operation the Parameter_Classification Algorithm will be used to classify the status of the parameters. After the classification of individual parameters the Network State Classification Algorithm will predict the network state. This state prediction will be used in different operations of the protocol as discussed later.

- *Training the Protocol has to be done by intentionally varying the levels of congestion and channel errors so as to artificially create network conditions similar to state00, state01, state10 and state11*
- *The parameters MRi, Ri and Si should be calculated and stored till the values become stable or the standard deviation of the calculated parameters are very less using* ***Parameter_Adaptation()***
- *Determination of Threshold values like {LMRi, HMRi} for MRi parameter,{LRi} for parameter Ri, and {LSi, HSi} for Si using Algorithm* ***Threshold_Calculation()***
- *For each Window after all the transmitted packets are ACKed, the MRi, Ri and Si parameters are updated and classify the Status of Parameters MRi, Si and Ri as Low, Medium or High using* ***Parameter_Classification()*** *Algorithm*
- *Classify the Network States through a hierarchal checking of the values MRi, Ri and Si according to Fig. 25 and using* ***Network_State_Classification()*** *Algorithm*

Fig. 27 Overall Operation of the Protocol

C. Training Phase of the Protocol

The input to the Parameter_ Adaptation algorithm as shown are the RTT values and the State to which the RTT values corresponds to and the output from this Algorithm are the updated internal Tables of the Protocol which contains the Minimum and Maximum value obtained by each of the parameter MRi, Ri and Si for all the states 00, 01, 10 and 11. The Algorithm in this case calculates the values of the parameters MRi, Ri and Si after each window and stores it until the values of the parameters become stable or their standard deviation is very small. After that only the minimum and maximum values are calculated. It has been seen from

Fig. 10 to Fig. 21 how the values become eventually stable. This is needed not to get locked to very high fluctuating values and the long term characteristics of the network can be ascertained.

```
Input: RTT values and State i
Output: Tables with Min and  Max values MRi, Ri and Si for all
states
Parameter_Adaptation ()
    /* storing of parameters till they are stable for the specified
   state */

while( (stddev(MRi) && stddev(Ri) && stddev(Si)) > 0)
   if ( All ACK Received) // After all transmitted packets are ACKed
        Set MRi by (5) and store as per state i;
        Set Ri by (9) and store as per state i;
        Set Si by Algorithm Stability () and store as per state i;
      end;
end;

   /* calculation of Min and Max of the parameters */
Calculate Min(MRi) and Max(MRi) and update Table as per state i;
Calculate Min(Ri) and Max(Ri) and update Table as per state i;
Calculate Min(Si) and Max(Si) and update Table as per state i;
end;
```

Fig. 28 Parameter Adaptation Algorithm

The input to the Threshold_Calculation Algorithm is the Tables from where the Algorithm outputs the threshold values using equations already discussed. The threshold values are stored internally and updated when the protocol is again run in the training mode to refine the threshold values as necessary.

```
Input: Tables with Min, Max values of MRi, Ri and Si for all States
Output: LMR, HMRi, LRi, LSi, HSi
Threshold_Calculation ()
  /* calculate thresholds from stored values in tables */
  LMRi = Max {Max (MRi) _State00, Max (MRi)_State01};
  HMRi = Max (MRi_State11);
  LRi = Max_Ri_State00;
  LSi = Max_Si_State10;
  HSi = Min{Min(Si)_State00, Min(Si)_State01};
  end;
```

Fig. 29 Threshold Calculation Algorithm

D. Operational Phase of the Protocol

The Parameter Classification algorithm shown in Fig. 30 classifies the status of the parameters MRi. Ri and Si into low denoted by L, medium denoted by M and high denoted by H. Actually each classification of parameters carries with it an estimation of the possible network state. For example a low value of MRi is possible in state 00 and 01 whereas a medium value is possible in state 10 and 11. A very high value is certainly seen in state10. This is because in state 00 and 01 the load on the network is not that high as compared to state 10, where only congestion and no error persists. The state 10 appears in medium case also because, when the degree of congestion is less or when during state 10 errors starts happening and the data transmission may be slowed down, the Mean Rise MRi may come in the medium zone. Similarly a low value of Ri can be only possible in state 00 and a high value can be possible in all the other states like 01,10 and 11. This is because both congestion and error in channel increase the standard deviation of the Mean RTT values. Again a very high stability can be only seen in state 00 and 01 where congestion is not there. The lowest stability is seen in the state 10 where the data transmission is maximum and stability the least. A medium value of stability can be seen in state 11 when both congestion and errors persists.

```
Input: MRi, Ri, Si
Output: MRi_Status, Ri_Status, Si_Status
Parameter_Classification ()
/* Updates status of parameters as low (L), medium (M) and high
(H) */
if(MRi <= LMRi)
MRi_Status = L;
elseif(MRi > LMRi && MRi <= HMRi)
MRi_Status = M;
elseif(MRi > HMRi)
MRi_Status = H;

if(Ri < LRi)
Si_Status = L;
else
Ri_Status = H;

if(Si < LSi)
Si_Status = L;
elseif(Si > LSi && Si < HSi)
Si_Status = M;
elseif(Si > HSi)
Si_Status = H;
end;
```

Fig. 30 Parameter Classification Algorithm

So it can be seen that considering a single parameter, the network can be estimated to be within a set of the network states. When all these individual estimations are combined the network state classification can be done using Network_Classification() algorithm shown in Fig. 31.

VI. DESIGN CONSIDERATIONS OF ADAPTIVE MULTI STATE PROACTIVE TCP

This section describes the internal detail of the proposed protocol and also discusses the rationale of the specific approach. This section describes the Simulation based experiments which have been carried out to analyze in detail how the different protocol related parameters respond to packet error rates with a view to enhance the performance of the protocol under high error conditions.

Open Science Index, Computer and Information Engineering Vol:3, No:4, 2009 publications.waset.org/916/pdf

Open Science Index, Computer and Information Engineering Vol:3, No:4, 2009 publications.waset.org/916/pdf

```
Input: MRi_Status, Ri_Status, Si_Status
Output: Network_State
Network_State_Classifiaction ()
if (MRi_Status = L)
    if (Ri_Status = L)
      if (Si_Status = H)
        State = 00;
      else
        State = Unclassified;
    else
      if (Si_Status = H)
        State = 01;
      else
        State = Unclassified;
end;
if (MRi_Status = M) {
    if (Ri_Status = H) {
      if(Si_Status = L)
          State = 10;
       else if(Si_Status = M)
          State = 11;
}
else
   State = Unclassified;
 return State;
end;
```

Fig. 31 Network State Classification Algorithm

A. Selective Acknowledgment Scheme

Whenever there is a loss due to duplicate acknowledgment or due to timeout, generally transport protocols transmit all the packets again starting from the point the highest acknowledgment was received if cumulative ACK scheme is used. When multiple packet losses appear in the same window the repeated transmission of packets already transmitted leads to some performance degradation. So in the Adaptive Multi State Proactive TCP Selective ACK scheme is used, where selectively packets can be retransmitted.

B. Effect of Packet Error on Protocol Parameters

After the implementation of the selective acknowledgment scheme in the Adaptive Multi-State Proactive TCP a simulation is carried out to understand the implication of high packet error rates on the protocol parameters using the simulation scenario described in Section VII with 10 senders communication to 10 receivers via a 5Mbps satellite link. The simulation is run for 500secs and the frequency of Mean RTT increase for three successive windows within that period, which generally calls for a proactive reduction of the congestion window, is observed. This frequency is an important parameter as with each such increase a preventive action is associated which sacrifices the bandwidth so as to increase the congestion epoch. The number times DUP ACK is received is also observed as with each such reception is associated a reduction of congestion window.

The event of a timeout expiry is the most important factor for a huge degradation in throughput because generally the congestion window is reduced to one and the retransmission mechanism in the event of a timeout sends all the packets again starting from the highest cumulative ACK received even if the Selective ACK option is used. So the performance of any protocol is highly dependent on the number of times timeout expiry is encountered in the total span of the connection. Generally Timeout phenomenon can happen because of two major reasons, first a huge congestion is developed in the network and the estimation of the time that the ACK should come or the Retransmission Timeout (RTO) estimated from previous experiences falls short and a timeout is experienced. This type of timeout expiry can be handled by increasing the RTO value but increasing the RTO value will lead to degradation in throughput in cases when the lack of arrival of ACK is not congestion. Second type of timeout events are seen when a very high PER is observed and in these cases the cause of timeout is high bit error rates and increasing the RTO value is not going to solve the frequency of timeout events.

In the simulation experiments conducted we have tried to ascertain the effect of Packet Error rates on the frequency of Timeout events. In the simulation experiment using scenario of Section VII, the Packet Error rates is gradually increased from a very low value of 0.0001 to very high value of 0.1 and the total number of Windows, RTT Increase Frequency, DUP ACK Frequency, Timeout Frequency and achieved throughput is plotted as shown in Table IV. The maximum bandwidth utilization possible for each connection is 500kbps if they are fairly [15] [14] sharing the 5Mbps of Satellite bandwidth available.

Fig. 32 shows the total number of windows or in other words rounds which could be completed within the 500 secs simulation time. It can be seen that as the packet error rate increases, due to more retransmissions the protocol is able to complete lesser number of rounds of data transfer. This parameter is more useful to visualize the amount of time the protocol keeps waiting because of an increase in the retransmission timer. In PER of 0.0001 within a 500 secs time, 842 rounds could be completed and in PER of 0.1 only 318 rounds could be completed. This shows why the throughput is degraded as the PER increases.

Fig. 33 shows the Frequency of three successive increase in mean RTT values within the simulation time which triggers a proactive correction of congestion window. As the PER increases this frequency is seen to reduce because with more errors the amount of data transmitted by the protocol also reduces. This frequency is important from the fact that with each such occurrence there is a proactive decrease of the congestion window. Another important fact which is observed is that, at very high error rates when the amount of data transmitted is reduced there is an appreciable amount of proactive reduction attributed by error initiated RTT increase.

Fig. 34 shows the frequency of DUP ACK reception with PER and it has been seen that initially the DUP ACK frequency increases with increasing PER and then it reduces since less data being transmitted because of frequent reduction of the congestion window during the DUP ACK handling.

Open Science Index, Computer and Information Engineering Vol:3, No:4, 2009 publications.waset.org/916/pdf

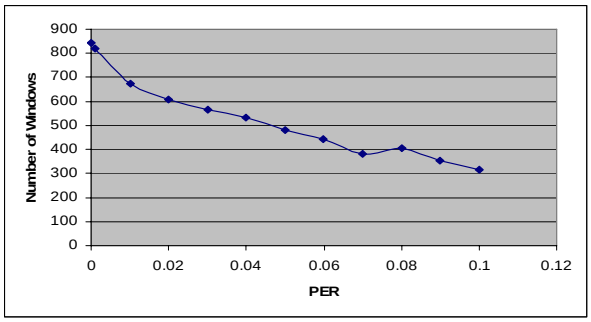


Fig. 32 Number of Windows vs PER

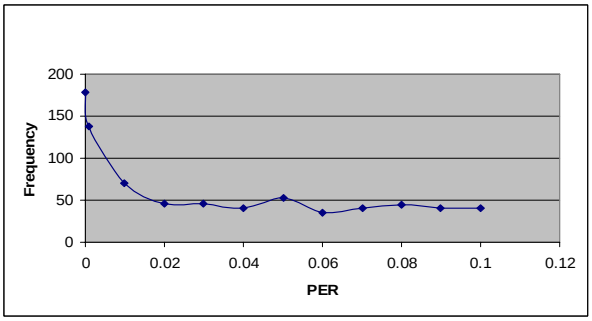


Fig. 33 RTT_Increase_Frequency with PER

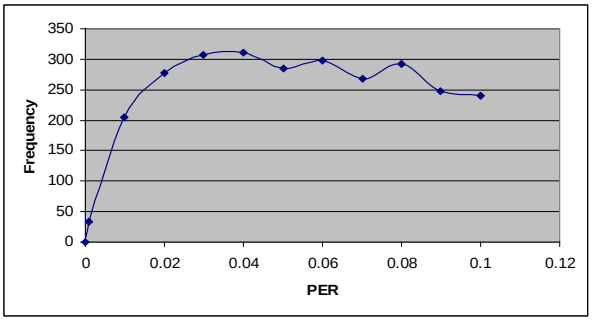


Fig. 34 Dup_ACK_Frequency

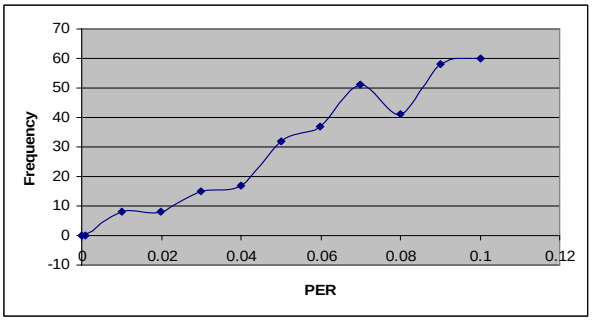


Fig. 35 Timeout_Frequency with PER

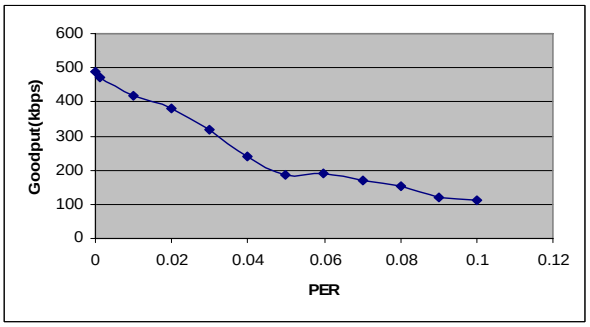


Fig. 36 Goodput (kbps) with PER

TABLE IV
SIMULATION RESULT FOR PROTOCOL PARAMETER VS PER

PER	*Total number of Windows*	*RTT Increase Freq*	*Dup ACK Freq*	*Timeout Freq*	*Through put (kbps)*
.0001	842	179	0	0	488.5
.001	819	138	33	0	473.0
.01	675	70	204	8	419.4
.02	609	46	277	8	381.1
.03	566	46	307	15	318.8
.04	533	41	311	17	240.7
.05	480	53	285	32	185.0
.06	445	35	298	37	192.3
.07	381	41	269	51	170.3
.08	403	45	293	41	151.3
.09	353	41	248	58	118.4
.1	318	40	241	60	111.2

Fig. 35 shows the frequency of occurrence of timeout along with PER. It can be seen that as the PER increases the frequency of timeout increases appreciably, with no timeout for PER of 0.001, 8 timeouts for PER of 0.01 and 60 timeouts for PER of 0.1. The timeouts are the most expensive events in terms of retransmission and increase of RTO value leads to the very high degradation of throughput.

Fig. 36 shows the achieved goodput with varying PER of the individual connection. The maximum possible bandwidth is 500kbps and it can be seen that 488.5kbps is achieved at PER of 0.0001 but the throughput drastically falls to 111.2 kbps at a PER of 0.1. From Fig. 32 to Fig. 36 the major factor which has to be considered in the design, for the performance improvement under high error condition has been analyzed so as to improve the performance of the protocol in high error cases. In Adaptive Multi-State Proactive TCP design the following new concepts have been used to enhance the performance.

C. State based Timeout Action

It has been observed that the RTT is not an independent random variable [9] and has correlation with congestion which is generally assumed but the correlation with error is not generally considered. In case of high error condition the RTT values are affected as discussed in Section II. So in high error cases a probability exists of mean RTT increase being signaled when the cause is error and not congestion. This leads to repetitive decrease of cwnd by the penalty factor and degradation of performance.

Another area of performance improvement is the modification of the Retransmission Timeout Algorithm. It is obvious that TCP protocol performance is highly dependent on the proper design of the Timeout algorithm. Generally the performance of almost all TCP Protocols degrade during a high congestion and high error condition. The major contributing factor for this huge performance degradation is the increase in the Retransmit Timer value. The protocol waits in the RTO loop much of the time which is advantageous if

Open Science Index, Computer and Information Engineering Vol:3, No:4, 2009 publications.waset.org/916/pdf

the cause of loss is congestion as the ACKs can be received after an increased interval of time. But in case of error the ACK is not going to come so waiting beyond a certain time will lead to a degradation of throughput.

Moreover because of high error it has been observed that the number of timeouts increase appreciably as shown in Fig. 35 and no design of Timeout algorithm can get rid of this timer expiry as packet are getting corrupted because of channel errors and not because of an overestimation of transmission rate. But traditionally after a timeout expiry the congestion window is reduced to one and Slowstart is applied to reach to the optimum network capacity. This creates a major degradation when timeouts are induced by channel errors. In Multi-State TCP a modification has been done and the congestion window will not be reduced to one in case of timeout event when the state of the network is 01 as shown in Fig. 37. The logic is that if we already know that the network is in a no congestion high error state there is no advantage in going for a reduction of congestion window. In other states like 00, 10 and 11 the normal timeout algorithm is used and with a timeout the congestion window is reduced to one and Proactive Slow Start begins. Because of this approach the moment the network changes state from 01 to other states like 00, 10 or 11 and there is a chance of getting congestion the timeout initiated congestion window reduction is retained. This logic prevents the Multi State TCP performance from getting too much degraded in case the classification logic misclassifies and proper action needed for controlling congestion is not taken. In that case the network will move to congested state and there a DUP ACK or timeout will be handled in the most cautious way by reducing the rate to almost zero and the network will be quickly freed from congestion.

```
State_based_Timeout_Action ()
if (RTO expiry)
        if (state = 01)
          MAXRTO = 2 * IRTT
          /* cwnd not changed */
         Proactive_Congestion_Avoidance ();
        end;
        elseif (state = 00| 10 | 11)
          cwnd = 1;
          Proactive_Slow_Start();
        end;
 end;
```

Fig. 37 State based Timeout Action

So in Multi State TCP we propose to use a modified Maximum Retransmission Timeout. Basically we want to keep the RTO values within limit and in error induced loss the binary bakeoff algorithm leads to severe performance reduction. In Multi State TCP during state 01 i.e. when no congestion and high error dominates the network the Max RTO will be kept to twice the Ideal RTT value. An ns2 based simulation has been carried out to find the optimal value of the maximum retransmission timeout so that the protocol gets a optimal throughput in all types of packet error rates.

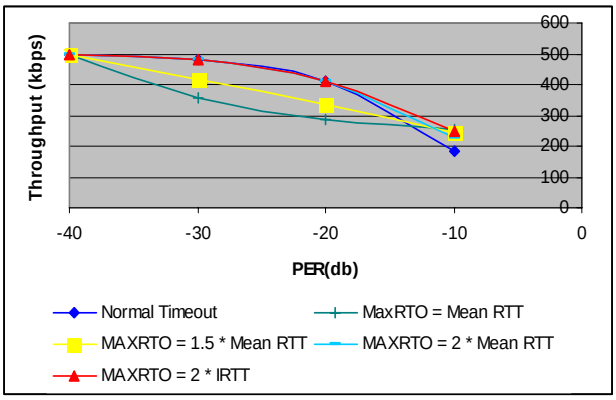


Fig. 38 Throughput vs. PER for values of MAXRTO

Fig. 38 shows the simulation result obtained by using different values of the MAXRTO. In the simulation for No Congestion and High Error state 01 this MAXRTO value is used and other states used the normal MAXRTO of upto six binary backoff leading to 64 times initial RTO. Since the protocol uses the normal Timeout mechanism for the other states only in very high PER of -10db an improvement can be seen when the MAXRTO is kept at twice the ideal RTT.

D. State based Error Recovery

Multi State TCP uses a state based error handling mechanism. The major changes in Multi State TCP are in state 01 where no reduction of congestion window is done in case of DUP ACK reception. Only the Fast Retransmit is done of the lost packet. The logic is that once it is known that the lost packet is because of error and not congestion the reduction in congestion window will only degrade the performance [23][24]. For state 00 reduction of congestion window is done to ¾ of its prevailing value instead to ½. The logic is already a proactive action is being taken periodically with mean rise for three successive windows so reduction of ¾ instead of ½ may enhance the performance specially when the state is estimated to have no congestion and error. In state 10 or 11 where the congestion is estimated to be present the reduction factor is kept at ½ so that the fairness property of the protocol is maintained under presence of congestion as shown in Fig. 39.

```
State_based_Error_Recovery ()
  if (state = 00)
    cwnd = cwnd * ¾
  elseif (state = 10 | 11)
    cwnd = cwnd * ½;
  /* for state 01 no reduction in cwnd */
    Fast_Retransmit (Lost_Segment);
end;
```

Fig. 39 State_based_Error_Recovery Algorithm

Open Science Index, Computer and Information Engineering Vol:3, No:4, 2009 publications.waset.org/916/pdf

E. Adaptive Penalty Factor

In the Multi- State TCP whenever an incipient congestion is anticipated by the successive increase in the mean RTT for three successive windows the proactive reduction of congestion window is undertaken which is the heart of the successful operation of the protocol. How much should be the penalty factor determines the tradeoff between precaution to congestion and compromise to throughput. There is a need to estimate the amount of extra data in the router buffer and to proactively adjust the congestion window to keep the extra data in the network within control. The Mean Rise MRi calculated from all the RTT samples basically denotes the percentage by which the network buffers are overloaded. So (1 – MRi) can be considered to be a very good estimate of the penalty factor which is multiplied with the congestion window on every three successive rise of Mean RTT.

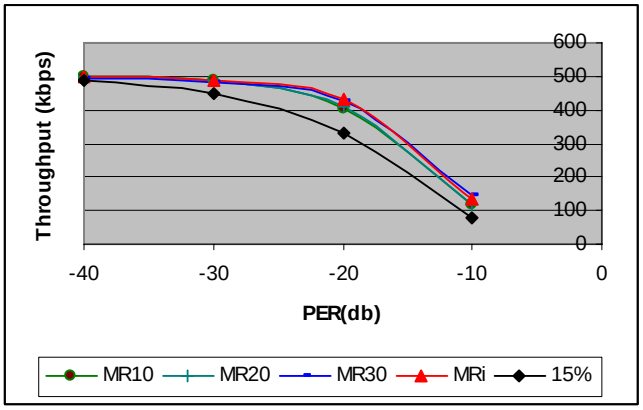


Fig. 40 Throughput vs PER for different Penalty Factor

In Fig. 40 a comparison has been made to see the impact of the choice of penalty factor on the throughput of the protocol for different values of PER. It has been seen that use of the different Mean Rise values gives a better throughput than the flat 15% obtained empirically through a series of simulations in ns2[28]. Out of all the Mean Rise values MRi gives the best throughput when used for the penalty factor considering all PER rates. This is because these parameters are adaptively obtained from the network and carries more information about the dynamic network condition.

F. State based Proactive Action

At very high channel error condition the probability of an error induced increase of RTT is more than congestion induced increase of RTT. So in state 01 there will be no proactive reduction of the congestion window with the penalty factor so that the congestion window is not reduced to a very low value. Moreover it has also been seen that probability of state transition from state11 to state 01 is more because high congestion high error state often leads to state 01, as whenever there is more errors in the channel timeouts are inevitable and that leads to a drastic reduction of the congestion window and hence the overall transmission rate leading to the state01. So, high congestion high error network states are also taken care with the modifications made in this algorithm.

```
State_based_Proactive_Action ()
  if(RTT_Increase_Detected)
    if(state = 00 | 10 | 11)
      cwnd = cwnd * (1 – MRi)
    /* for state 01 no change in cwnd */
end;
```

Fig. 41 State based Proactive Action Algorithm

VII. Test and Evaluation

We evaluate the performance of the proposed TCP in terms of goodput through simulations when several connections share the same link. We simulate the system as shown in the Fig. 42 below 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. The segments may get lost with a packet error rate PER. 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 = 500s which is 1000 times the round trip time value.

A. Goodput Calculation

Goodput is the effective amount of data delivered through the network. It is a direct indicator of network performance. We expect that a good TCP scheme transmit as much data as possible, while behaving friendly to other TCP flows in terms of consuming the network resource. In the following graphs shown the throughput of the protocol is compared to TCP SACK [6] and TCP Vegas [5] in the same test bed of 10 senders communicating to 10 receivers using a 5Mb bottleneck link via satellite. To artificially induce congestion the bandwidth between Earth Station A and B is stepwise reduced to see the reaction of the protocol to congestion. Here all ten different connections are connected to the Earth Station with 500kbps bandwidth each. So the minimum aggregate bandwidth required for all the connections to perform

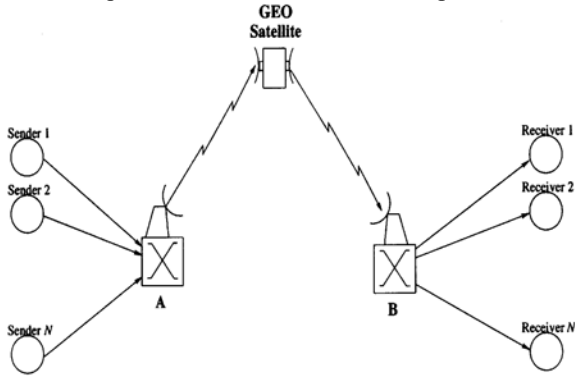


Fig. 42 Simulation Scenario

optimally is 5Mbps. Table V shows the goodput of the Multi State TCP along with Vegas [5], TCP SACK [6] and Proactive TCP [28]. In Table VI the percentage utilization is shown and it can be seen that Multi State TCP utilizes 50% bandwidth even in a very high PER of 0.1. Table VII shows the appreciation over its peers and in high error condition the appreciation is 600% over Vegas and 800% over SACK. This is a very significant improvement achieved with following graphs showing the result in Fig. 43, Fig. 44 and Fig. 45.

TABLE V
GOODPUT FOR CONGESTED NETWORK WITH PER

PER	Vegas (kbps)	SACK (kbps)	Proactive TCP (kbps)	MultiState TCP (kbps)
0.001 (-30db)	244.6	225	450	481
0.01 (-20 db)	168.7	116	323.6	408.5
0.1 (-10 db)	34.9	27	86.67	244.9

TABLE VI
PERCENTAGE BANDWIDTH UTILIZATION

PER	Vegas (%)	SACK (%)	Proactive TCP (%)	MultiState TCP (%)
0.001 (-30db)	48.92	45.00	90.00	96.20
0.01 (-20 db)	33.74	23.20	64.7	81.70
0.1 (-10 db)	6.98	5.40	17.33	48.99

TABLE VII
PERCENTAGE APPRECIATION OF MULTI-STATE TCP OVER ITS PEERS FOR DIFFERENT PER

PER	Vegas (%)	SACK (%)	Proactive TCP (%)
0.001 (-30db)	96.64	113.77	6.88
0.01 (-20 db)	142.14	252.15	26.23
0.1 (-10 db)	**601.97**	**807.37**	**182.66**

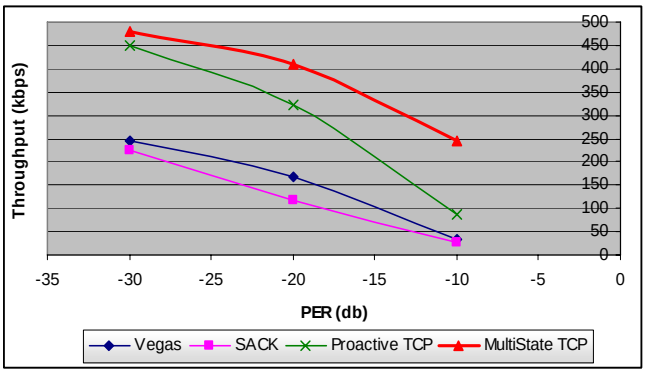


Fig. 43 Throughput for Congested Network with different PER

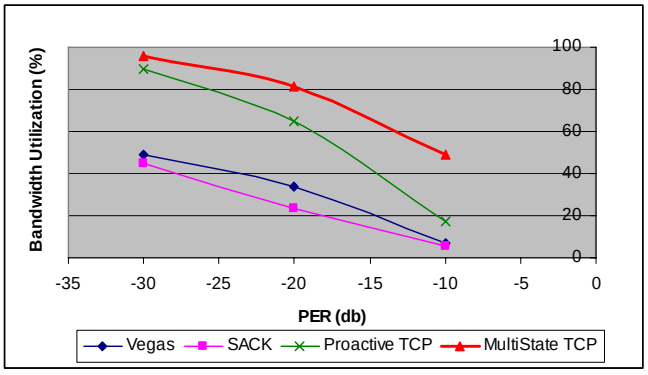


Fig. 44 Percentage Bandwidth Utilization

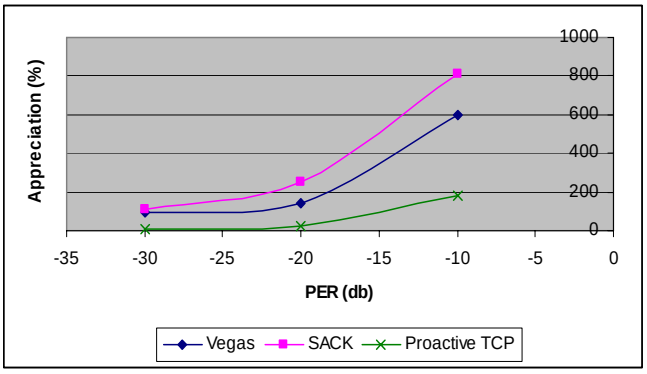


Fig. 45 Multi-State TCP Appreciation for different PER

VIII. Conclusion

This paper through detailed simulation and analysis of TCP over Satellite Networks has established the fact that multi state network classification is essential to enhance the overall throughput of the network which is affected equally by congestion and channel noise. The modeling and analysis as discussed in this paper has proved that the conventional two state network classification devised in classical TCP over wired network is not optimal for satellite based TCP protocol. So to enhance the performance of TCP protocols over Satellite based Networks there is a need for a Multi State representation of network condition by a combination of the Multi State Network Classification model to be developed which has been explained in this paper. For the design of such a model simulation based experiments are carried out to analyze the impact of congestion and error on RTT values. The inferences drawn from the analysis have been used to evolve a novel Statistical RTT based model which uses a hierarchical decision making process for network state classification. This model has been utilized in the design of an Adaptive Multi State Proactive Transport Protocol for performance enhancement of satellite based networks. The major causes of performance degradation have been analyzed and the novel schemes for throughput enhancement have been proposed. The design consideration of the protocol has been discussed thoroughly in this paper and it has been shown how an adaptive mechanism of learning the statistical parameters of a network can help in getting a significant performance benefit. The Multi–State TCP is seen to outperform substantially over its peers mainly in high error conditions. The most important merit is that the performance improvement obtained in this protocol [19] does not need any change in the Routers and all changes are restricted to the sender and receiver protocol stack [9].

Acknowledgment

Authors thank all members of Department of Computer Engineering, I.T. BHU, and all staff members of Advanced Communication Technology Division (ACTD), SAC/ISRO for their cooperation and technical support.

Open Science Index, Computer and Information Engineering Vol:3, No:4, 2009 publications.waset.org/916/pdf

Open Science Index, Computer and Information Engineering Vol:3, No:4, 2009 publications.waset.org/916/pdf

REFERENCES

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

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

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

[4] Jian Fang and Özgür B. Akan*,, "Performance of Multimedia Rate Control Protocols in InterPlaNetary Internet*" in IEEE COMMUNICATIONS LETTERS, VOL. 8, NO. 8, AUGUST 2004

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

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

[7] Bonald, T., "*Comparison of TCP Reno and TCP Vegas: Efficiency and Fairness*" in Proceedings of PERFORMANCE 99, Istanbul, Turkey, October 1999

[8] C. Casetti, M. Gerla, S. Mascolo, M.Y. Sanadidi, R. Wang, "*TCP Westwood: bandwidth estimation for enhanced transport over wireless links*" in Proceedings of Mobicom 2001, Rome, Italy July (2001)

[9] J. Martin, A. Nilsson, and I. Rhee, "*Delay-Based Congestion Avoidance for TCP*" in IEEE/ACM Transactions on Networking, vol. 11, no. 3, pp. 356–369, June 2003

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

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

[12] P. Sinha, N. Venkitaraman, R. Sivakumar, V. Bharghavan, "*WTCP: A Reliable Transport Protocol for Wireless Wide-Area Networks*" in Proc.ACM MOBICOM 1999, pp. 231-241, Seattle, Washington, August 1999

[13] D. T. Tran, F. J. Lawas-Grodek, R. P. Dimond, W. D. Ivancic, "*SCPSTP TCP and Rate-Based Protocol Evaluation for High-Delay, Error-Prone Links*" in *SpaceOps 2002*, Houston, TX, October 2002

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

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

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

[17] Consultative Committee for Space Data Systems, "*Space Communications Protocol Specification-Transport Protocol (SCPS-TP)*" in Recommendation for Space Data Systems Standards, CCSDS 714.0-B-1., Blue Book. Issue 1, Washington, D.C.: CCSDS, May 1999.

[18] T.V.Lakshman and U.Madhow, "*The performance of TCP/IP for networks with high bandwidth-delay products and random loss*" in *IEEE/ACM Trans. Networking*, vol. 5, June 1997.

[19] Injong Rhee and Lisong Xu, "*Limitations of Equation-based Congestion Control"* in *IEEE/ACM Trans. Networking*, June 2007

[20] C. Metz, "*TCP over satellite the final frontier*" in IEEE Internet Compuingting., pp. 76–80, Jan. /Feb. 1999.

[21] C. Partridge and T. J. Shepard, "*TCP/IP performance over satellite links*" in IEEE Network Magazine., pp. 44–49, Sept-Oct. 1997.

[22] M. Allman, "*Ongoing TCP research related to satellites*" in RFC 2760, Feb. 2000.

[23] W. Stevens, "*TCP/IP Illustrated"* Addison-Wesley, 1994, vol.1

[24] R. Fox, "*TCP big window and NAC options*", RFC 106, IETF, June 1989.

[25] H. Bal Krishnan, S. Seshan, E. Amir and R. H. Katz, "*Improving TCP/IP performance over wireless networks*" in Proc. ACM Mobicom, Nov. 1995, pp. 2–15.

[26] S. Dawkins, G. Montenegro, M. Kojo, V. Magret, and N. Vaidya, "End *to-end performance implications of links with errors*", Internet draft, Mar. 2000.

[27] Ravi.S.Prasad, Manish Jain and Constantinos Dovrolis, "*On the Effectiveness of Delay-Based Congestion Avoidance* "

[28] Mohanchur Sarkar, K,K,Shukla, K.S.Dasgupta, "*A Proactive Transport Protocol for Performance Enhancement of Satellite based Networks*" in (IJFCA) International Journal of Futuristic Computer Applications (Accepted)

[29] A. A. Awadallah and C. Rai, "*TCP-BFA: Buffer Fill Avoidance*" in *Proc. IFIP High Performance Networking Conf.*, Sept. 1998, pp. 575–594.

Mohanchur Sarkar received his B.Tech degree in Computer Science and Technology from Kalyani University, India in 1999. He joined Space Applications Centre a branch of the Indian Space Research Organisation (ISRO) in 2000. Since then he has been working on projects related to Satellite communication. His research interest include protocols related to Satellite based Networks, Satellite based Navigation, Delay Tolerant Networks, ,DVB-RCS, DVB-S and DVB-SH. Presently he is working as Scientist/Engineer–SE in the Advanced Communication Technology Group of ISRO and also pursuing Ph.D. in Computer Science and Engineering from I.T,BHU.

Dr.K.K.Shukla did his B.E., M.Tech. and Ph.D. degrees from I.T.BHU. Presently he is a Professor (Ex-Head) in the Department of Computer Engineering, I.T.B.H.U. He has contributed significantly in the field of Artificial Intelligence, Neural Networks, Information Security, Computer Networks, and Computer Graphics. He has published a huge number of papers in National and International Journals and Conferences.

Dr.K.S.Dasgupta received his B.E and M.E degrees in Electronics and Telecommunication from Jadavpur University, India in 1972 and 1973. He received his Ph.D. degrees in Computer Science from Indian Institute of Technology (IIT) Mumbai in 1990. Currently he is the Deputy Director of SatCom and I.T Application Area of Space Application Centre (ISRO) and Director of Development & Educational Communication Unit (ISRO). Dr. K S Dasgupta has joined Space Applications Centre (ISRO) in 1974. Since then he has contributed significantly in the field of Image Processing and Satellite Communications. He is actively involved in various design and development of ground based and space borne communication systems in the Centre. Dr. Dasgupta's contributions are significant as he has developed many new techniques and approaches in the field of Digital Signal Processing, Digital Image Processing, Computer Architecture and Digital Communication and Onboard Processing Satellites. He is the chief technical architect of the GSAT-4 regenerative payload to be launched this year and heading the satellite based navigation and distance education programmes of ISRO.