# A Two State Proactive Transport Protocol for Satellite based Networks

MOHANCHUR SARKAR
Satellite Communication and Navigation Application Area (SNAA)
Space Application Centre, ISRO
Ahmedabad, Gujarat
INDIA
msarkar@sac.isro.gov.in | mohanchur@yahoo.com

K.K.SHUKLA
Department of Computer Engineering
Institute of Technology, BHU
Varanasi
INDIA
kkshukla.cse@itbhu.ac.in

K.S.DASGUPTA
Indian Institute of Space Science and Technology (IIST)
Thiruvananthapuram
INDIA
ksd@iist.ac.in

*Abstract:* - In satellite-based networks, current TCP Protocols have lower throughput performance mainly due to the effect of long propagation delays and high link error rates. In this paper, a new congestion control protocol for satellite-based networks is proposed. The protocol uses a proactive approach and is composed of novel ideas like Proactive Slow Start, Proactive Congestion Avoidance and Decision based Error handling policies that are combined with traditional TCP algorithms, like Fast Retransmit. The mainstay of our protocol is the nature of the RTT pattern can give us indication of an incipient congestion in the network. This changing pattern of RTT is used to differentiate between congestion and link error, thus avoiding unnecessary rate throttle. In the initial phase, necessary augmentation of ns2 simulator pertaining to the proposed protocol has been carried out. This was essential to create a necessary test bed for exhaustive simulation of the protocol considering a GEO network, with different congestion level and packet error rate. Simulation results show that the protocol always outperforms other TCP protocols in terms of goodput and an improvement of 80% to 120% is observed, especially when the packet error rate is very high. Evaluation of the protocol shows a high fairness property and excellent adaptability to high levels of congestion and channel errors.

*Key-Words:* - TCP, Satellite, UDP, Peach, Reno, SACK, Vegas

## 1. Introduction

Over the years TCP has become the de-facto protocol standard for congestion control in the existing terrestrial Internet. However, analytical and experimental studies [9] confirm that in general the current TCP protocols variants have performance problems in networks with long propagation delays, and comparatively high link error rates such as satellite networks [10], [13], [14]. The throughput of any TCP connection is generally found to be reciprocal to the round-trip time (RTT) of a connection, and is directly proportional to the congestion window (cwnd), which represents the amount of unacknowledged data the sender can have in transit to the receiver. TCP throughput decreases in a satellite based network mainly because [10], [13], [14] the long propagation delays cause longer duration of the *Slow Start* phase during which the sender may not use the available bandwidth optimally. The problem of improving TCP over satellite has been widely investigated in the literature [10], [11], [12].

### 1.1 Major problems in a Satellite based Network

One of the major problems in a satellite-based network is the random packet errors, which are not common in the wired counterpart. TCP protocols react to the lack of arrival of acknowledgements or duplicate ACK as a sign of congestion. Therefore, 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 [9], Peach+, TP-Planet [7], and RCS [5]. Other approaches to transport protocol design are found in [1], [3], [16], [18], and [19] where the transport protocol stack in the sender and receiver are only modified. Transport protocols with network-assisted operation are given in [4].

### 1.2 Basic Approaches to Congestion Control

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. It increases until the point when Duplicate ACK 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 an aim to maximize the throughput always drive the network to the maximum capacity after which every connection suffers the collateral damage caused by the overestimation of the channel capacity. All the AIMD TCP protocol variants, like Tahoe [19], Reno, New Reno, SACK [17], Peach [9], Peach+, and TP-Planet [7] fall in this category. Reactive algorithms try to solve the problem but do not 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 [2][3]. TCP Vegas [18] is a proactive protocol, which is also recommended by the CCSDS SCPS-TP [4] for use in satellite-based networks. The performance of AIMD protocols in satellite networks in discussed in [6].

In this paper, we introduce a new congestion control scheme for satellite-based networks, which is an end-to-end solution to improve the throughput performance in satellite-based networks. The paper is organized as follows. We introduce the new TCP Protocol in Section 2. In Section 3, we evaluate the performance of the proposed protocol through simulation. Finally, in Section 4 we conclude the paper.

## 2. Proactive Congestion Control

The Transport Protocol proposed in this paper uses a proactive approach to the problem of handling congestion in the network. In this section the main philosophy of proactive congestion control, simulations pertaining to the proactive approach, the main actions taken in the proactive congestion control along with its main features and design issues are discussed.

### 2.1 Main Philosophy of Proactive Congestion Control

The main philosophy of this work is that congestion in a network does not grow all of a sudden. The queues in the routers start growing, and that leads to more packet delay or an increase in the experienced RTT. The nature of the pattern of the RTT can give us some indication of the incipient congestion in the network. There exists some empirical value or limits on the RTT which can signal an incipient congestion. If this changing pattern of RTT is used to take decision in whether a packet loss is because of congestion or link error, then the unnecessary rate throttle need not be done.

### 2.2 Simulations for the Proactive Approach

We have carried out simulation to verify the impact of congestion and channel error on the RTT experienced by a connection. First, the network is kept within is allowable limit and connections are allowed with congestion window that the network can very well handle, and the pattern of RTT is observed. It is seen that the mean RTT remains within a limit. Then the network is slowly congested by increasing the congestion window of the prevailing connections, keeping packet error rates to zero. It is observed that in majority of the cases an increase in mean RTT occurs before a congestive loss is encountered. Then the connections are kept within the capacity and packet error rates are introduced. In this case duplicate ACK start coming but the mean RTT is found to be within a limit. This confirmed that the pattern of the RTT could be used to take decisions, in a proactive way to avoid congestion and handle errors.

## 2.3 Main Action of the Proactive Approach

The protocol proposed in this paper 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 windows, then it can be anticipated that the network is getting in to the congested state. Here, the decision-making criterion is that how many congestion windows 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. However, it may lead to a point where the network is so much congested that even taking corrective measures does not help. Thus, 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 giving direction and the next increase will confirm that detection so that corrective action can be taken. If decision is taken just by considering two windows then that may lead to false decisions. We have considered taking the increase in mean RTT rather than considering individual RTT. This is because of the fact that the variation in individual RTT may be attributed by other dynamic network factors. For example, the choice of the different optimized path by the router and processing delays that are temporary and taking decision on that basis creates more oscillations of the congestion window. However, when a successive increase in the mean RTT is observed it signals of some major change happening in the network condition.

## 2.4 Determination of Penalty Factor

After an incipient congestion is detected, the congestion window is reduced by a penalty factor. The choice of the penalty factor signifying the amount by which the window should be decreased is very important for the throughput of the protocol. The value should be chosen such 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 with different value of the penalty factor from 0.1 to 1.0 as shown in Fig. 1. Penalty factor of 0.1 means a 90% decrease of the congestion window and 1.0 signifies no change to the window thereby indicating not using this logic of proactive congestion avoidance algorithm. The simulation of throughput with different values of penalty factor is plotted in Fig. 1 and it is seen that with penalty factor 1.0 the throughput degrades drastically, which shows the justification of the proactive approach. A peak in the throughput is at 0.85 and 0.90, which corresponds to 15% to 10% decrease. Thus, the protocol uses a penalty factor of 0.85 with the detection of incipient congestion thereby removing the extra amount of data getting into the network. This will try to keep the overall network load within the tolerable limit avoiding a congestive loss and an eventual congestive meltdown of the network. The value of the penalty factor will ultimately decide the effectiveness of the proposed protocol.

## 2.5 Proactive Slow Start

In the beginning of a new connection, the sender executes the Slow Start algorithm to probe the availability of bandwidth along the path. During the slow start phase, the window doubles itself every RTT and quickly captures the channel bandwidth until the slow start threshold is reached. After that, the congestion window increases linearly. The slow start phase creates performance degradation in cases the RTT is high. Peach [8], Peach+ use dummy packets to acquire the available capacity within two RTT by a technique named emulated slowstart. This is an efficient technique but requires special capability of the routers to discard low priority packets. Moreover, during time of congestion this will add to the wastage of bandwidth. Vegas [18] uses a more conservative approach by increasing the congestion window every alternate RTT to avoid overestimation of network capacity. In the proposed protocol, as the principle is a slow and steady rise of the congestion window, we propose to retain the old slow start mechanism. The difference is that Proactive Slow Start phase will consider the incipient congestion algorithm. Before doubling its congestion window, it will check if an incipient congestion is detected by an increase in the mean RTT.

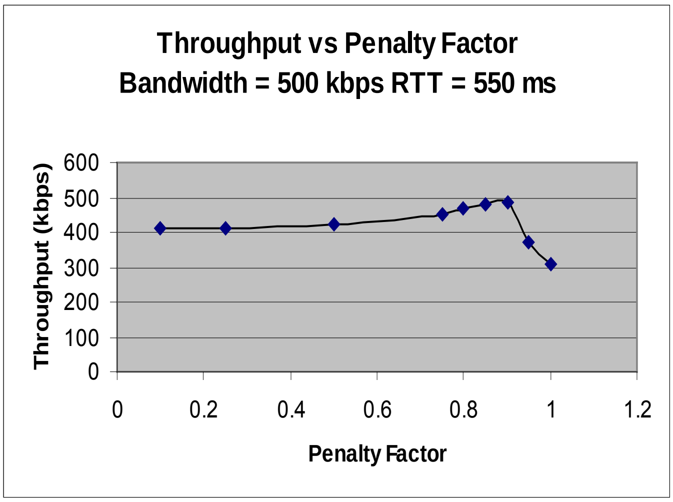


Fig.1 Evaluation of Throughput for different Penalty Factor

In that case, the congestion window will not be doubled but rather decreased by the penalty factor. This will lead to a very controlled way of capturing the network capacity. When a connection starts, if the network is not congested slow start will perform normally, but if congestion exists in the network the new connection will not pump too much packet in the network. The logic behind the proactive approach is that the collateral damage has to be avoided. If the network is already getting into congested state a new connection in the process of getting too much resource, should not create performance degradation to other connections in the network.

## 2.6 Decision Based Error Recovery

TCP was initially developed for wire-line networks where the link error rate is low such that the majority of the segment losses are due to network congestions. Thus, the sender assumes that all segment losses are caused by congestions and accordingly it decreases its transmission rate. Although the application of forward error correction code can increase the reliability of satellite links, satellite networks have several orders of magnitude higher error rates than the wire-line networks [14]. As a result, we cannot ignore the errors in satellite links and assume that all segment losses occur due to congestion. This assumption may lead to drastic and unnecessary decrease in resource utilization [14], [17].

### 2.6.1 Loss Detection Algorithm

This problem could be solved if TCP could have been able to distinguish whether segment losses occur due to network congestion or due to link errors [13]. However, this is currently infeasible; in [14] the authors suggest decoupling error and congestion control. TCP would then be responsible only for congestion control while the link layer handles the error control. However, this solution is impractical because the link layers of all sub networks composing the Internet needs to be redesigned. An alternative solution is that the sender could contain an algorithm, which can distinguish between congestion and error initiated losses often termed as Loss Detection Algorithm.

However, such an algorithm must be very reliable. In fact, if this algorithm does not respond correctly to actual network congestion the network utilization decreases drastically [14]. To our knowledge, such a reliable algorithm does not exist to date. Peach [9], Peach+, TP-Planet [7] distinguishes errors and congestion by physically probing the network to confirm whether there is capacity in the network by sending low priority dummy packets and monitoring the reception of their acknowledgement. These dummy packets adds a high overhead of 17% [9] on the satellite channel and requires special capability of the routers to drop low priority packets. This leads to wastage of the precious satellite bandwidth and its deployment requires a change in routers along the path.

### 2.6.2 Two State Proactive Approach

The proposed protocol is considered moving between two broad states: An incipient-congestion state, which is signaled by the increase in RTT and an un-congested state. This state information is used by the protocol to handle the losses due to channel errors. Thus, whenever three duplicate ACKs are received the protocol will check whether it is in an incipient-congestion state or not. If yes, the congestion window and slow start threshold will be reduced to half. However, if the protocol is in the un-congested state the congestion window and slow start threshold will be kept unchanged only a Fast Retransmit will be done for the packet lost. Then the congestion window can increase linearly with each received ACK. In the start of a connection, the protocol is in the un-congested state and in this state if mean RTT is not increased for three successive windows the protocol remains in this state.

Whenever an increase in mean RTT is perceived for three consecutive windows, the state of the protocol changes to the incipient-congestion state. In the incipient-congestion state, again if increase in mean RTT is seen then the protocol remains in that state. When a decrease of mean RTT is observed for three successive windows the protocol moves to the un-congested state as seen in Fig.2 else it remains in the incipient-congestion state. The logic is that in an incipient-congestion state the probability of a congestive loss is more. On the other hand, in the un-congested state the probability of loss due to packet error is more. This logic is used in the Decision Based Error Recovery mechanism.

## 2.7 Overall Working of Proactive TCP

In Fig.3, the overall mechanism of the proposed Proactive TCP scheme is depicted. 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 un-congested state. The receiver window signifies the maximum number of segments the receiver can accommodate. The value of the receiver window is obtained during the SYN message exchange. 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. However, the mean RTT is also checked during this phase and if RTT increase is detected for three successive congestion windows, the congestion window is reduced by the penalty factor.

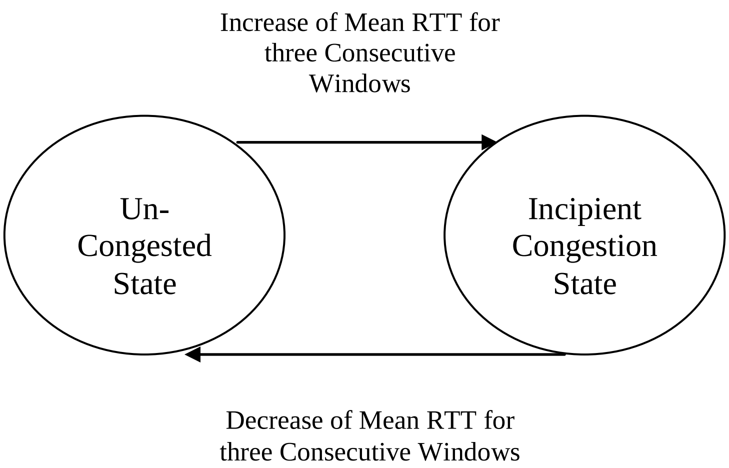


Fig.2 State Transition Diagram

### 2.7.1 Proactive Congestion Avoidance

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. If an acknowledgement is not received for a transmitted segment within its retransmission timeout period, the timer expires and the Proactive Slow Start phase is again initiated. During the Proactive Congestion Avoidance phase, the mean RTT is checked every time at the end of a round of data transfer. If an increase in RTT for three successive congestion windows is detected, the congestion window is reduced by the penalty factor and the state of the protocol goes to incipient congestion state. Now if three duplicate ACK are received the protocol calls the Fast Retransmit algorithm, a retransmission of the lost segment is done and the protocol moves to the Decision based Error Recovery Phase.

In this phase if the protocol is in an incipient-congestion state, which signals that the probability of the network to be in a congested state is more, and a packet is not received by the receiver. Hence, chance of the packet being lost by congestion is more likely than the chance that the packet was lost because of channel error. Moreover, it has to be considered that even though the Proactive Congestion Avoidance algorithm has already taken action by periodic reduction of congestion window, to avoid congestion in the network, the segment loss is detected. Therefore, the congestion window is reduced to half and protocol moves to the Proactive Congestion avoidance phase. If the state of the protocol is un-congested state, then no reduction in congestion window is done and protocol moves to Proactive Congestion avoidance as it is anticipated to be a loss due to error in the channel. The state of the protocol is changed from incipient congestion state to un-congested state when a decrease in RTT for three successive congestion windows is observed. This shows the network is in an uncongested state.

**2.7.2 Proactive Timeout Action**

In case, that a segment is retransmitted by Fast Retransmit but again it gets lost or its acknowledgement gets lost because of either congestion or corruption the retransmission timer will expire which signals a major problem in the network. In this case, the congestion window is reduced to one and slow start threshold reduced to half of its previous value. Proactive Slow Start is initiated, which will increase the congestion window in a fast but conservative manner to adapt to the available capacity.

In this protocol, the timeout algorithm is not changed; the binary back off algorithm [19] is used which doubles the timeout value with every timeout expiry. The timeout algorithm is kept unchanged keeping in view of its stable and proven nature and in case that there in a false alarm in the decision making process, the congestion window will be reduced with the timeout expiry. This will keep the network capacity under control in cases of heavy congestion or high error condition.

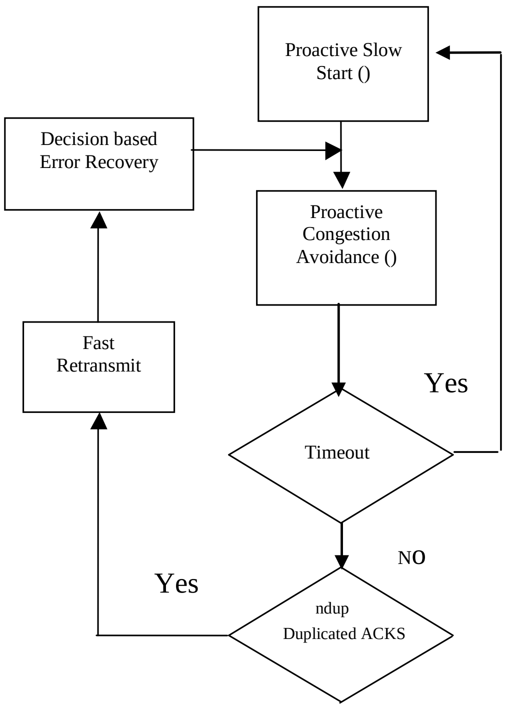


Fig.3 Flow Chart of Proactive TCP

## 3. SIMULATIONS AND ANALYSIS

The performance of the proposed proactive transport protocol is evaluated in terms of goodput and fairness in a simulation environment where several connections share the same link. Fig.4 shows the simulation scenario where 10 senders transmit data to 10 receivers through a satellite channel. The 10 streams are multiplexed in Earth Station A, whose buffer can accommodate 25 segments. The segments may get lost with a packet error rate (PER). In this experiment, all the 10 senders are each connected to the Earth station A, with a link of bandwidth 500kbps and RTT of 10ms. All the 10 receivers are connected to Earth station B with a 500kbps link with RTT 10ms. The receiver window (rwnd) is set to 64 segments, the link between Earth Station A to B via satellite to be 5Mb and the RTT between the two stations as 550ms. The time of simulation have been kept to 550 seconds, which is 1000 times the round trip time.

Through the simulation-based experiments, the characteristics of the protocol will be ascertained. Any transport protocol will have dependence with respect to changing RTT. The behavior of the protocol in terms of changing channel condition has also to be ascertained. The other important aspect of any transport protocol is the ability of the protocol to handle different degree of congestion. One of the most important properties of a transport protocol is its ability to fairly share the available bandwidth and allow the coexistence of other flows in the shared medium [20].

In the following sections, simulation-based experiments are discussed for characterizing protocol performance and finding the (i) Protocol dependence on varying RTT, (ii) Protocol dependence on varying PER, (iii) Protocol dependence on varying congestion levels, (iv) Congestion Window Evolution of the protocol and, (v) Fairness property of the protocol.

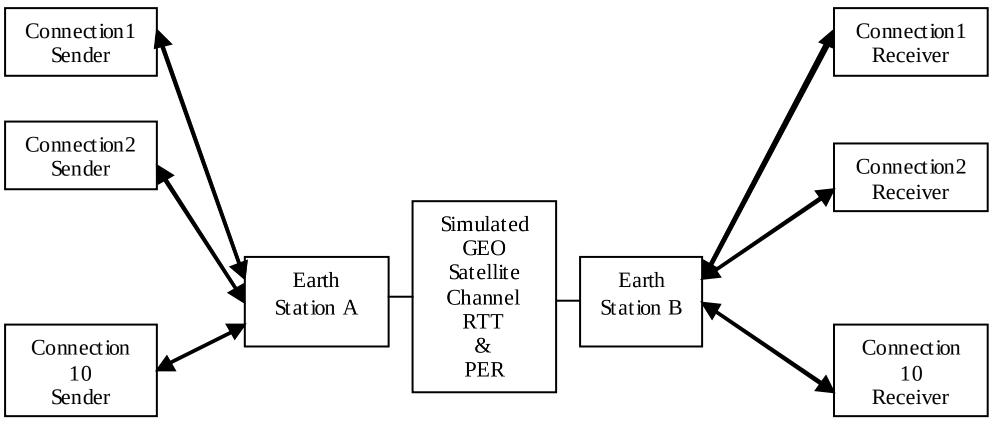


Fig.4 Simulation Scenario

### 3.1 Protocol Dependence on RTT

Generally, the performance of all transport protocols degrades with increasing RTT values. The performance achieved in case of terrestrial Internet connections, where RTT ranges to a few milliseconds is much better than what is obtained in a satellite based network. Moreover, the performance achieved in LEO based satellite network is better than obtained in MEO based Networks. Thus, the performance of any transport protocol highly depends on the type of network in which it will be used. In the following subsection, the performance of the protocol with increasing RTT values has been investigated. The simulation experiments also consider different packet error rate, as any protocol performance is dependent on the channel error also.

All the ten connections share a common satellite channel between Station A and B with a link capacity of 5Mbps. The delay between the Station A and B is varied from very low RTT values, typically seen in LEO satellite networks to high RTT values experienced in MEO networks. In Fig.5, the throughput obtained by the individual connections for increasing RTT is plotted. The plot corresponds also to different PER values ranging from $10^{-3}$ to $10^{-1}$. It can be seen that, the throughput degrades with increasing values of RTT. It can also be seen that, when the RTT is small even a high PER does not significantly degrade the throughput. In the left hand part of Fig.5, it can be seen that the curves for different PER are closer but with high RTT, the curves are all highly dependent on the PER. Another significant advantage of the protocol

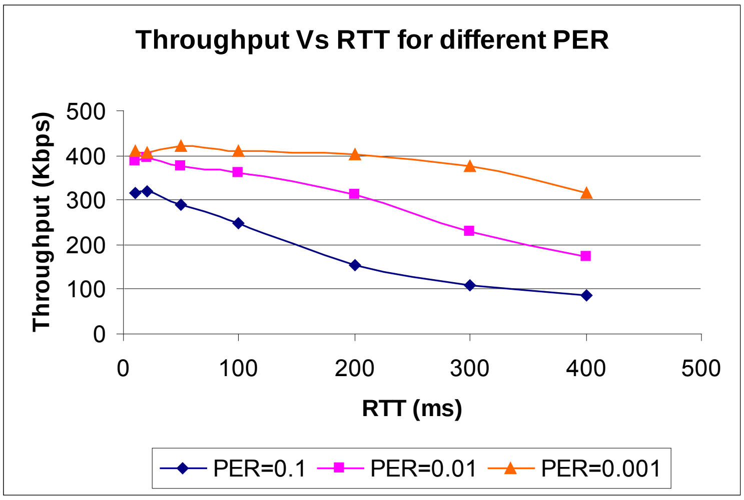


Fig.5 New TCP Throughput for varying RTT for different PER

is that, even with high error rates of $10^{-2}$ and $10^{-1}$ the throughput is not degraded drastically. From Fig.5, it can be seen that when the PER is less like $10^{-3}$, the performance of the protocol degrades very slowly with increasing RTT values. Therefore, the protocol handles the issue of high RTT quite well.

### 3.2 Protocol dependence on PER

To provide reliability to the transmitted data, all transport protocols follow the Automatic Repeat Request (ARQ) mechanism by which packets once corrupted or dropped are retransmitted again. This repeated transmission consumes bandwidth and extra time of the sender. All transport protocol performance is highly dependent on the packet error rates in the channel. In this section, the dependence of the protocol in terms of increasing packet error rates has been considered. The Packet Error Rate of the link between Earth Station A to B is varied from $10^{-4}$ to $10^{-1}$ in steps of 0.01. The mean throughput obtained by all the 10 connections is used in generating the graph. A very high throughput is obtained when the PER is $10^{-3}$ or less. From the graph in Fig.6, it can be seen that the throughput is reasonably good is case of PER less than $10^{-2}$.

From Fig.6, it can be seen that the protocol performance does not degrade drastically because of the Decision based error handling technique, which restricts the fall of the congestion window when error induced losses are encountered. The reduction of congestion window because of error is a major factor for performance degradation in satellite channel.

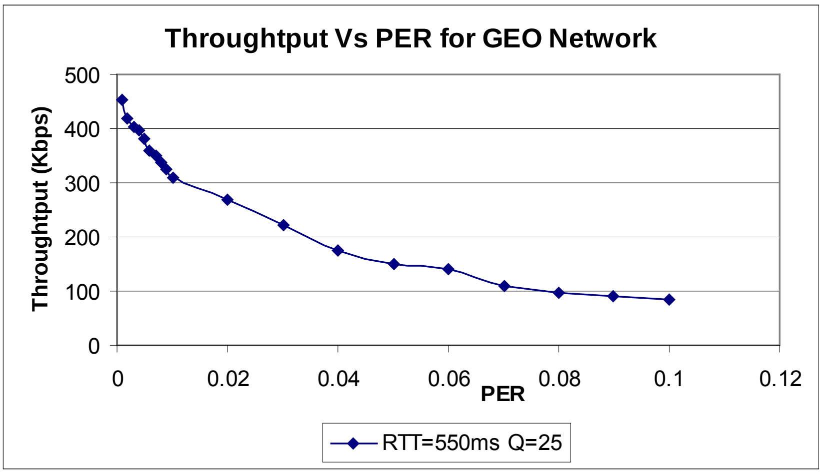


Fig.6 New TCP Throughput for increasing PER

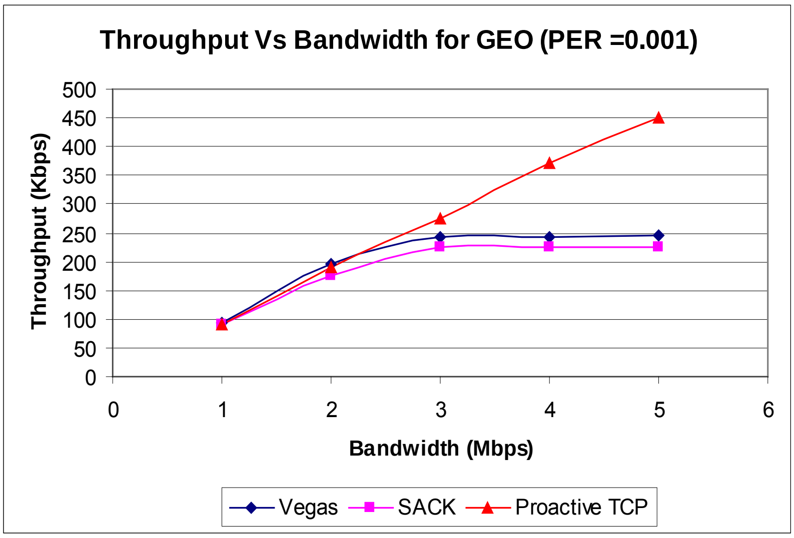


Fig.7 Throughput with Bandwidth PER = 0.001

### 3.3 Protocol Dependence on varying Congestion Levels

Goodput is the effective amount of data delivered through the network and is a direct indicator of network performance. It is expected 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 e.g. bandwidth. In the simulation experiment, the throughput of the protocol is compared to TCP SACK [16] and TCP Vegas [18], in the same test bed of 10 senders communicating to 10 receivers using a 5Mbps bottleneck link via satellite. The throughput achieved by all the individual connections is averaged to generate the graphs in Fig.7, Fig.8, and Fig.9.

To induce higher levels of congestion intentionally the bandwidth between Earth Station A and B is stepwise reduced to see the reaction of the protocol to congestion. All ten different senders are connected to the Earth Station with 500kbps bandwidth. Therefore, the minimum aggregate bandwidth required for all the connections to perform optimally is 5Mbps. When we reduce the bandwidth on the satellite link, this will lead to congestion in the network and it is the job of the protocol to handle the congestion by reducing the transmission rate and adapt to the changing network condition and available bandwidth.

In the first case, a PER of $10^{-3}$ is assumed and it can be seen from Fig.7 that at 5Mbps bandwidth, which signifies the minimum capacity needed the proposed TCP outperforms TCP SACK and TCP Vegas. In Fig.7, it can also be seen than when the congestion is very high as 1Mbps bottleneck link the protocol performs smoothly. It gets a throughput close to 100 kbps so the effect of congestion has drastically been reduced. Moreover, from Fig.7 it can also be observed that as we go on reducing the bandwidth, the protocol adapts to the new available capacity outperforming SACK and Vegas. This is possible because of the proactive nature of the protocol, which detects an incipient congestion very quickly and takes corrective measures.

The effect of packet error rate is very well handled by the decision based error recovery technique as it can be seen that the protocol gets a very good utilization of the available capacity. The utilization is close to 90%, in this case and this shows the efficiency of the proactive approach to congestion control. Thus, Proactive TCP is seen to adapt well to the higher levels of congestion. When the satellite-link bandwidth is 5Mbps, it gets a 450kbps goodput, where the maximum capacity for individual connections is 500 kbps. As 10 connections share the link, when the bottleneck satellite bandwidth reduces from 5Mbps to 4Mbps, 3Mbps, 2Mbps and 1Mbps, the individual available capacity translates from 500kbps to 400kbps, 300kbps, 200kbps to 100kbps. From Fig.7, it can be observed that the goodput of individual connections are always close to these values in each of the cases.

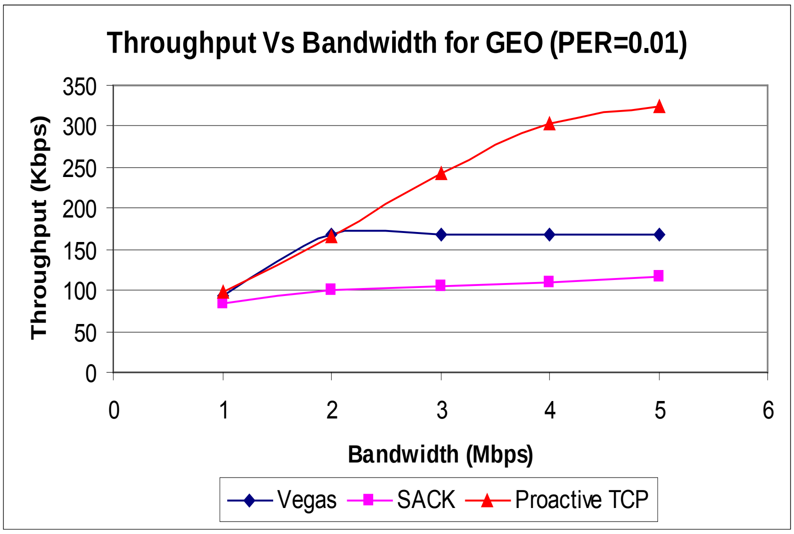


Fig.8 Throughput with Bandwidth PER = 0.01

In Fig.8, the performance of the protocol under a high packet error rate of $10^{-2}$ is considered. This is a high packet error rate and it can be seen that under such a high error condition the other transport protocols like SACK and Vegas are not properly able to use the available capacity. This is because, these protocols are not able to differentiate the cause of loss due to congestion and channel error.

Therefore, the prevailing channel error rate in the satellite link dominates the throughput, as with each duplicate ACK received the congestion window is reduced by these protocols. This has been validated through these simulations.

The Proactive TCP on the other hand well adapts to the increasing congestion levels and available capacity and gets a good share of the bandwidth. Moreover, the effect of packet errors is also not that predominant. As seen in Fig.8, in this case of higher packet error rate goodput is close to the available capacity but lower than what is seen in the Fig.7, where packet error rate has been lower.

At low congestion levels close to 70% goodput is achieved as seen in Fig.8. In this case, also a good adaptation to higher congestion levels is exhibited by the protocol.

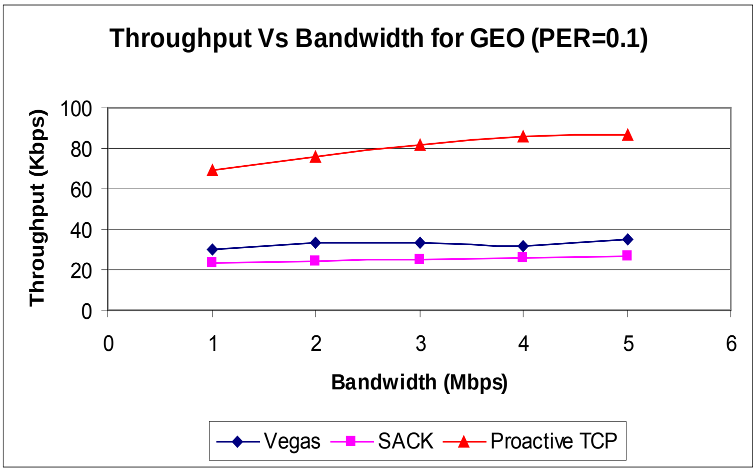


Fig.9 Throughput with Bandwidth PER = 0.1

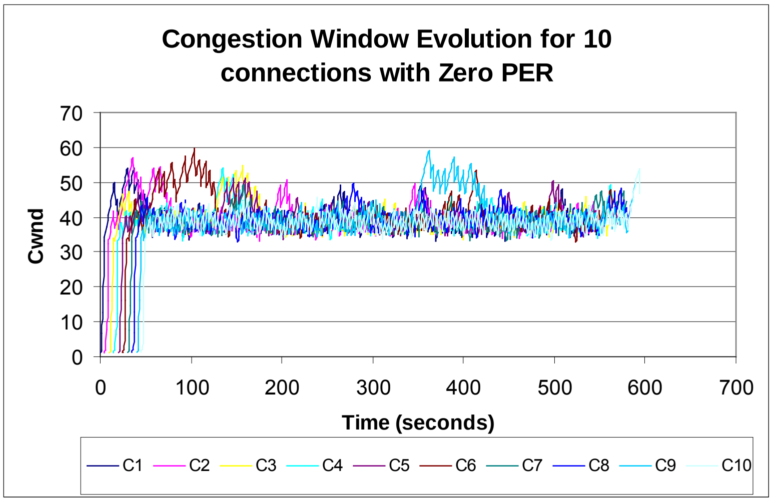


Fig.10 Congestion Window Variation of all 10 connections for GEO Network at zero PER

The simulation results in Fig.9, considers the case of very high packet error rate of $10^{-1}$. It can be seen that the performance of TCP SACK and Vegas is drastically reduced in this case.

However, it can be seen that Proactive TCP outperforms SACK [16] and Vegas [18] by more than 100% in this condition. This is because of the fact that the protocol uses the prediction logic for differentiating congestive and corruptive losses. The major advantage of the proactive approach to congestion control and the decision based error recovery technique can be appreciated in this case

## 3.4 Impact on the Congestion Window

The total time between two successive congestion events is called the congestion epoch having significant impact on the throughput of a connection. The larger the congestion epoch better is the utilization of the channel capacity and the capability of the protocol in handling congestion. The main intent of the Proactive Transport Protocol is to handle congestion and device a mechanism by which the connections can self regulate the rate, and the overall rate of all the connections prevailing in the network can be kept below the maximum allowable capacity. In this section, through simulation-based experiments it is verified whether the intent is achieved by checking the congestion window evolution of the connections. In Fig.10 the congestion window evolution for all the 10 different connections is shown. In this case, a hypothetical, packet error rate of zero has been considered. The intent is to see, how well the proposed protocol handles the congestion in the network. From Fig.10, it can be seen that the congestion window for all the connections, progresses and self regulates their value by proactively sensing the overestimation. The important point to note is that the congestion window for all the connections does not drastically fall to lower values. This shows that the congestion epoch has been increased and the congestive meltdown of the network does not happen in this case. This has been the main motive of the design of the proactive transport protocol.

In Fig.11, the congestion window evolution for all the 10 connections are shown, and it can be seen that, all the connections fairly maintain their transmission rate even in case of errors. However, reduction of congestion window to half is seen and in some cases when some connections face the timeout expiry the congestion window is reduced to one. Nevertheless, there is no major congestive meltdown of the network and all the connections fairly regulate their transmission rate.

The small oscillation of the congestion window typically when the window value reaches the maximum capacity allowable, which is the novel characteristics of the proactive approach to congestion control always keeps the network immune form any major catastrophic overestimation of network capacity.

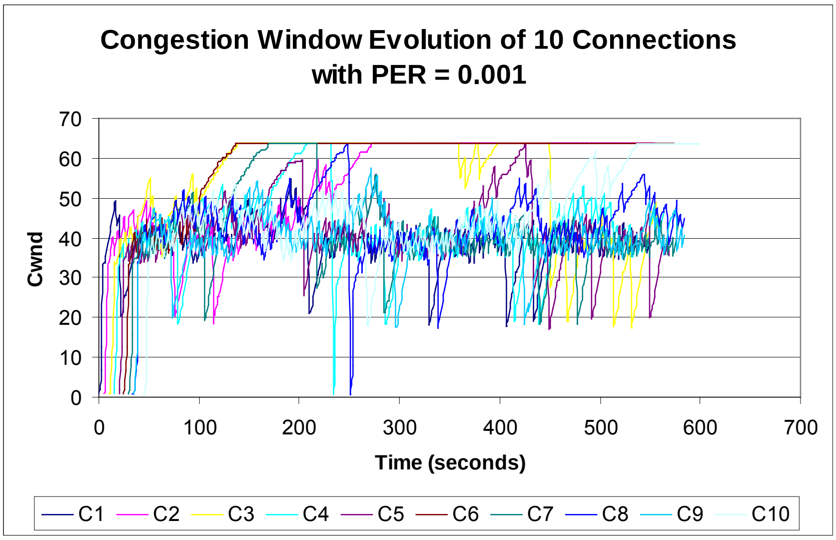


Fig.11 Congestion Window Variation for 10 connections GEO Network for PER 0.001

### 3.5 Fairness of Proposed TCP

The fairness of a protocol is the ability of the connections to share the available bandwidth. This is needed to ensure that it does not happen that the connection that starts first takes an appreciable proportion of the bandwidth, and does not release the capacity even when other connections are active in the network. This property is very much desirable in any new protocol and in the process of enhancement of throughput; the fairness property should not be violated [9]. This is especially needed for connections to be used over a shared medium. Thus, multiple connections of the same TCP scheme must interoperate nicely and converge to their fair share. The fairness index function (1) is proposed in [20], to evaluate the fairness of TCP schemes. The fairness index function is expressed as:

$$F(x) = \frac{(\sum_{i=1}^{N} x_i)^2}{N * (\sum_{i=1}^{N} x^2{}_i)} \qquad (1)$$

Where xi is the throughput of the ith connection and N is the number of connections. F (x) ranges from 1/N to 1.0. A perfectly fair bandwidth allocation would result in a fairness index of 1.0. On the contrary, if all bandwidth were consumed by one connection, (1) would yield 1/N.

| | **5 Mb** | **4Mb** | **3Mb** | **2 Mb** | **1 Mb** |
|---|---|---|---|---|---|
| **000** | .999720 | .998838 | .996617 | .991967 | .988334 |
| **.001** | .999182 | .998104 | .996283 | .966535 | .970683 |
| **.01** | .993726 | .993611 | .996394 | .995048 | .986218 |
| **.1** | .999995 | .998715 | .992898 | .994957 | .988345 |

Table 1 Jain’s Fairness Index for different Bandwidth and Packet Error rates

By undergoing the fairness analysis, it is ascertained that the protocol can be used in a shared medium. This is more important as in any network having a decentralized infrastructure with no proper regulation of bandwidth; prevailing connections should not let other connections starve because they enter the network late. From Table5 it can be seen that all the connections get an almost equal share of bandwidth [20].

In Table1, the columns correspond to decreasing levels of bandwidth of the satellite link, which will create increasing congestion in the routers, as the data received in an Earth Station is not getting the needed bandwidth to deliver the data to the destination. The rows correspond to increasing rates of packet error starting from no error to a PER of 0.1 where 1 out of 10 packets are corrupted. The protocol is seen to maintain a high degree of fairness. Even during high degree of congestion and error, only a very slight degradation of fairness with increasing congestion or packet error can be seen.

### 3.6 Simulation with increased Connections and Bandwidth

In this section, a simulation scenario similar to Fig.4 with 42 nodes is considered to analyze the performance of the protocol for increasing number of connections. In Table 2, the throughput obtained under different link capacity of the satellite channel with 20 active connections is shown after being averaged. The percentage utilization is also shown in Table 2 considering the achieved throughput, number of connection and bandwidth of the channel. This shows that the proactive protocol performance does not degrade with increasing number of connections.

| **Bandwidth (Mbps)** | **10.0** | **5.0** | **4.0** | **3.0** | **2.0** | **1.0** |
|---|---|---|---|---|---|---|
| **Throughput (kbps)** | 493.770 | 246.356 | 196.738 | 145.954 | 97.830 | 49.315 |
| **Percentage Utilization (%)** | 98.754 | 98.542 | 98.369 | 97.302 | 97.830 | 98.630 |

Table 2 Throughput and Percentage Utilization with increased connections and bandwidth

## 4. Conclusion

The behaviour of RTT in normal network condition and a congested network condition is evaluated. As discussed in this paper, we proposed a new TCP scheme to improve the performance of TCP in Satellite based networks and validated it through exhaustive simulations. The proactive TCP Protocol proposed is found to outperform its peers like Vegas and SACK under different levels of congestion and channel errors. The proactive nature of the protocol makes it quite robust in handling high degree of congestion, where the protocol is found to adapt well to the available capacity. The advantage obtained by using Proactive Slow Start is that connections initiated during congested network conditions do not cause collateral damage to other ongoing connections. The Decision based error recovery is very effective, in not letting the congestion window degrade by differentiating errors and congestion making the protocol suitable for use in satellite based networks. Simulation results show that the protocol always outperforms other TCP variants in terms of goodput by 80% to 120%, especially when the packet error rate is high. The significant merit is that the performance improvement obtained in this protocol does not need any change in the Routers and all changes are restricted to the sender and receiver protocol stack. Moreover, there is no extra overhead associated with the protocol for taking the decisions. The protocol is found to have fairness property as analyzed in the paper. Though we have evaluated the protocol for GEO satellite network, the protocol will also perform well in LEO or MEO based satellite networks. It can also be used in the terrestrial Internet for its good proactive congestion control and slow start mechanism.

## 5. Acknowledgment

Authors would like to thank Mr.Deval Mehta, Head SCTD, Mr. A.P.Shukla, Group Head, DCTG, and Mr.K.S.Parikh, Deputy Director, SNAA of Space Applications Centre (ISRO) for their constant encouragement towards the realization of this work.

## 6. References

[1] Mohanchur Sarkar, K.K.Shukla and K.S.Dasgupta, Network State Classification based on the Statistical Properties of RTT for an Adaptive Multi State Proactive Transport Protocol for Satellite based Networks, International Journal of Computer Networks and Communication (IJCNC), Vol. 2, No. 6, November 2010, pp. 155-174.

[2] Mohanchur Sarkar, K.K.Shukla, and K.S.Dasgupta, Performance Analysis of Proactive and Reactive Transport Protocols using a Proactive Congestion Avoidance Model, International Journal of Computer Applications (IJCA), Vol. 6, No. 5, September 2010, pp. 10-17.
[3] Mohanchur Sarkar, K.K.Shukla, and K.S.Dasgupta, A Proactive Transport Protocol for Performance Enhancement of Satellite based Network, International Journal of Computer Applications (IJCA), Vol. 1, No. 16, Feb 2010, pp. 114-121.
[4] Space Communications Protocol Specification (SCPS) - Transport Protocol (SCPS-TP), Recommended Standard CCSDS 714.0-B-2 Blue Book, October 2006.
[5] I. F. Akyildiz, O. B. Akan, and G. Morabito, A Rate Control Scheme for Adaptive Real-Time Applications in IP Networks with lossy links and long round trip times, IEEE/ACM Transactions on Networking, Vol. 13, Issue 3, June 2005, pp. 554-567.
[6] C. Liu and E. Modiano, On the performance of additive increase multiplicative decrease (AIMD) protocols in hybrid space terrestrial networks, Elsevier Computer Networks, Vol. 47, Issue 5, April 2005, pp. 661–678.
[7] O. B. Akan, J.Fang and I. F. Akyildiz, TP-Planet: A Reliable Transport Protocol for InterPlaNetary Internet, IEEE/SAC, Vol. 22, No. 2, February 2004, pp 348-61.
[8] Ian F. Akyildiz, Xin Zhang and Jian Fang, TCP-Peach+: Enhancement of TCP-Peach for Satellite IP Networks, IEEE COMMUNICATIONS LETTERS, Vol. 6, No. 7, July 2002.
[9] Ian F. Akyildiz, Giacomo Morabito and Sergio Palazzo, TCP-Peach: A New Congestion Control Scheme for Satellite IP Networks, IEEE/ACM TRANSACTIONS ON NETWORKING, Vol. 9, No. 3, June 2001, pp. 307-321.
[10] M. Allman, S.Dawkinks, et al., Ongoing TCP research related to satellites, RFC 2760, Feb 2000.
[11] C. Barakat, E. Altman, and W. Dabbous, On TCP performance in a heterogeneous network: a survey, IEEE Communications Magazine, Vol. 38, Issue 1, Jan 2000, pp. 40-46.
[12] N. Ghani and S. Dixit, TCP/IP enhancement for satellite networks, IEEE Communication Magazine, Vol. 37, No. 7, July 1999, pp. 64-72.
[13] C. Metz, TCP over satellite the final frontier, IEEE Internet Computing, Vol. 3, Issue 1, January 1999, pp. 76–80.
[14] C. Partridge and T. J. Shepard, TCP/IP performance over satellite links, IEEE Network Magazine, Vol 11, Issue 5, October 1997, pp. 44–49.
[15] T.V.Lakshman and U.Madhow, The performance of TCP/IP for networks with high bandwidth-delay products and random loss, IEEE/ACM Trans. Networking, Vol. 5, Issue 3, June 1997, pp. 336-350.
[16] Mathis, M., J. Mahdavi, S.Floyd, and A.Romanow, TCP Selective Acknowledgment Options, RFC 2018, April 1996.
[17] H. Balakrishnan, S. Seshan, E. Amir and R. H. Katz, Improving TCP/IP Performance over Wireless Networks, Proc. ACM MOBICOM, November 1995, pp. 2-11.
[18] L. S. Brakmo, S. O'Malley and L. L. Peterson, TCP Vegas: New Techniques for Congestion Detection and Avoidance, ACM SIGCOMM Computer Communication Review, Vol. 24, Issue 4, October 1994, pp. 24-35.
[19] V. Jacobson, Congestion avoidance and control , ACM SIGCOMM Computer Communication Review, Vol. 18, Issue 4, August 1988, pp. 314-329.
[20] R. Jain, D. Chiu and W.Hawe, A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems, DEC, Research Report TR-301, September1984.
[21] 21UCB/LBNL/VINT Network Simulator. http://www.isi.edu/nsnam/ns/

**Dr. Mohanchur Sarkar** received his B.Tech degree in Computer Science and Technology from Kalyani University, India in 1999 and Ph.D. Degree in Computer Science and Technology from I.T, BHU, in 2012.

He joined Space Applications Centre a branch of the Indian Space Research Organisation (ISRO) in 2000 and presently working as Scientist/Engineer-SE. He has been involved in several projects of national importance like EDUSAT R& D programme, Indian Regional Navigational Satellite system, Mobile Satellite Communication etc. He has to his credit many publications in reputed journals/conferences.

**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 Ph.D. degrees in Computer Science from Indian Institute of Technology (IIT) Mumbai in 1990. Presently he is the Director of the Indian Institute of Space Science and Technology (IIST), and former Deputy Director of SatCom and I.T Application Area of Space Application Centre (ISRO) and former 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 has been the chief technical architect of the GSAT-4 regenerative payload and headed satellite based distance education program and various projects and R&D activities of national importance of ISRO.