

Satellite-TCP: A Flow Control Algorithm for Satellite Network

K. Sindhu Raj^{1*}, B. Thanudas², N. Radhika¹ and V. S. Smitha³

¹Amrita Vishwa Vidyapeetham, Amrita Nagar PO, Coimbatore – 641 112, Tamil Nadu, India;
Sindhurajk805@gmail.com, n_radhika@cb.amrita.edu

²CNG, VSSC, ISRO, Thiruvananthapuram – 695022, Kerala, India; b_thanudas@vssc.gov.in

³VSSC, ISRO, Thiruvananthapuram – 695022, Kerala, India; vs_smitha@vssc.gov.in

Abstract

Objective: In this world of advanced technology, satellite communication is the basis for all data communications. TCP was designed to be flexible and robust but it cannot perform efficiently over the satellite links with long delay. The major objective of our work is to propose a novel flow control algorithm for satellite network to prevent the heavy degradation in the network performance due to long propagation delay, high bit error rate and bandwidth asymmetry. **Methods:** A strong flow control mechanism is necessary to eliminate the network performance degradation due to the satellite link characteristics like long propagation delay, high bit error rate and bandwidth asymmetry. Satellite-TCP is a flow control mechanism based on the negative acknowledgement strategy and connection splitting. The satellite link utilization is monitored and the active connections are assigned with the window values. The window values were considered as it can easily reflect the available bandwidth in the satellite links. **Findings:** The major problems affecting the performance in satellite links are long propagation delay, high bit error rate and bandwidth asymmetry. The proposed Flow Control Algorithm called Satellite-TCP is based on controlling the flow of Negative Acknowledgement packets (NACK), according to the level of congestion at the network. The two main modules in the proposed system include an enhanced TCP flow control mechanism and a data loss recovery mechanism. Among this the flow control is based on link utilization measurements and throughput calculation techniques. Data loss recovery strategy is based on Negative Acknowledgements coming from the receiver side. In this way, we can decrease the amount of packet loss in the network and even eliminate the packet loss, sometimes. Satellite-TCP adopts a window growth strategy to accelerate the speed of increase in the size of congestion window. The flow control mechanism not only enhanced the throughput of forward link but also greatly reduced the bandwidth occupancy rate in back-ward link in the satellite network. This in turn increases the overall performance of the satellite link. **Improvements:** If we add a priority based strategy to the new flow control mechanism, it can judge the network performance according to the data loss. Moreover we are expecting that, if we add a priority bit based strategy to the new flow control mechanism, it will help to distinguish the specific reasons for data loss and can take corresponding flow control strategies. While considering the throughput and loss rate, Satellite –TCP is showing excellent performance in the case of forward link but it performs poor in the case of backward link. So we are planning to extend our work in such a way that it will perform well in the case of backward link also.

Keywords: Bandwidth Asymmetry, Bit Error Rate, Flow Control Mechanism, Loss Recovery System, Propagation Delay, Satellite –TCP

*Author for correspondence

1. Introduction

Transmission Control Protocol (TCP) is a reliable protocol for communication between hosts. Most of the existing transmission control mechanisms are based on packet loss for congestion detection in the network. These existing methods are having a lot of drawbacks. This paper proposes a new novel flow control mechanism for the satellite network.

The communication through satellites provides global connections at any time. They play important roles during natural disasters, as they can reach remote and inaccessible areas at any time. Satellite link characteristics, namely long propagation delays, large bandwidth-delay products, and high bit error rates, have serious impact on the performance of satellite network. These traits may affect the data transmission adversely. In the case of satellite network, paths with large bandwidth-delay products, TCP needs a considerable amount of time to set its congestion window "CWND" to the appropriate value. Furthermore, TCP reacts to segment drops by lowering the congestion window value. When drops are caused by transmission errors, TCP unnecessarily reduces its transmission rate. Several methods are proposed to overcome these problems but most of them deploy end-to-end mechanism. Our aim is to implement the new Flow Control Algorithm for satellite network by adding a new data control strategy in between the sending and receiving gateways in the network protocol which is suitable for satellite network based on the traits of the satellite network. A window based flow control mechanism and a negative acknowledgement based loss recovery strategy act as the backbone of the flow control mechanism. The window size can be used for calculating the maximum amount of data in the communication pipe. We use acknowledgement packets, especially negative acknowledgements, for controlling the flow because the acknowledgement packets do not carry any data. They need very less amount of buffer space at the router as compared to the data packets. The basic components that measure the performance of satellite system are: throughput, bandwidth and transmission delay. The new system can be implemented in such a way that it should improve the above parameters so that the overall performance can be improved. The new strategy increases the congestion window size rapidly at the time of connection establishment. This helps to reduce the wastage of time for connection establishment as in the case of slow start strategy in the traditional TCP. It

aims at increasing the performance of communication over single-hop satellite links. Its flow control is based on link utilization measurements, and segment loss is handled with Negative Acknowledgements (NACKs). The simulation and analysis proved that the new flow control algorithm can increase the overall throughput of the satellite network and showed better performance as compared to existing flow control mechanisms.

2. Improving TCP/IP over Satellite Network

Recently, the research works in TCP introduced a lot of new methods for controlling the flow mechanism in TCP. Introduced a Random Early Detection and its advanced form FRED. It tries to prevent the congestion in the network by dropping packets before the router buffer get completely exhausted. This prevents timeouts. RED uses randomization to ensure same loss rate at all connections. When the average number of packets queued at the buffer exceeds a pre-defined threshold, the RED gateway starts to drop packets. Here the network connection with largest input rate will have the largest dropping percentage. FRED is an advanced version of RED. It generates a selective feedback to a selected set of network connections which have a large number of packets to get queued. This can be applied in wired network but these methods create unfair differences between the flows and create problems if applied in satellite networking. Explicit Congestion Notification (ECN) uses the TOS bits in the IP header to record congestion. Receivers use the next acknowledgement to inform the sender about the congestion. ECN uses two bits from the TCP header for the communication. One bit is used by the receiver to inform the sender about the congestion and the second bit is used by the sender to inform the receiver that it received the congestion information. ECN can be used in wired and wireless networks. The use of ECN in satellite network can be seen in³ and it proposed a new algorithm based on TCP (TCP-ATCA) that reduces the problems like long propagation delay and high bit error rate. This method is a combination of dynamic RED and ECN. This method reduces the transmission delay to a great extent. The main disadvantage of this method is that it needs modification to the existing TCP/IP protocol and also creates some disturbances in the flows⁴. Introduces a new algorithm to improve TCP throughput without modi-

ifying TCP protocol suit in satellite network. Since TCP uses the acknowledgments (ACKs) to adjust the sending rate, the basic idea for this scheme is to delay ACKs traveling through a node where its forward connection is congested. The algorithm aims to fully eliminate packet loss caused by buffer overflow. This scheme is implemented in the intermediate nodes connecting satellite links. The problem with this method is that, in delayed acknowledgement the delay for some interval for the acknowledgement packets should be specified by the users. Normally when the delayed acknowledgements are used, the Round Trip Time (RTT) estimated at the sender includes this delay also (since the sender just sees the RTT). So it causes performance delay⁵. Introduced an enhanced TCP-Vegas protocol for improving the TCP performance over satellite link. This increased the overall network utilization and throughput. TCP-Vegas perform poor in GEO satellite networks. So the new TCP Vegas-W introduces an improved Vegas protocol. But the problem here is that, it is based on the slow start mechanisms so we can't increase the increment speed of congestion window. Along with this the two major approaches in TCP performance enhancement are end-system policies and network policies. Congestion avoidance and data recovery methods come under end-system policies. Network policies handle resource utilization mechanisms.

2.1 End-system Policies

TCP considers any packet loss as an indication of network congestion and reduced its transmission rate to control the network flow⁶. The two main state variables in TCP are Congestion Window (CWND), which shows the number of unacknowledged outstanding packets, and Slow Start Threshold (SSTRESH), which identifies the point at which the function increasing CWND switches from exponential to linear behaviour. Currently, a number of flavours of TCP are existing. Among them the four major ways of TCP congestion control and avoidance are Vanilla TCP, TCP Reno, TCP New Reno, and TCP-Peach⁷. Vanilla TCP is one of the important flavours of TCP. Vanilla TCP includes both slow start and congestion avoidance algorithms. The network congestion is detected based on the retransmission timer. First SSTHRESH is set to the value of CWND then the slow start phase is applied to increment the CWND value. When CWND equals the value of SSTHRESH, the congestion avoidance phase gets activated. TCP Reno is a modified TCP

that includes fast retransmit and recovery algorithm. Any duplicate acknowledgement from the receiver side indicates a packet loss. The sender TCP retransmits the lost packet immediately after getting three duplicate ACKs. After this, the fast recovery algorithm makes $CWND = CWND/2$ and then increases it by one for every duplicate ACK. After this, TCP enters the congestion avoidance phase. This makes it possible to recover the lost packet within one RTT. The pitfalls in TCP Reno are cleared by TCP New Reno⁶. TCP New Reno avoids multiple window reductions in one window of data and considers the burstiness of the sender upon leaving fast recovery. TCP SACK is another TCP flavour. The receiver provides correct information about the properly received data packets using SACK TCP. All the lost data packets are retransmitted during the retransmission phase and then only the new data packets get transmitted. Here the lost packet recovery occurs within one RTT. Vanilla TCP performs efficiently in many situations but is not optimal networks with long RTT as lost packets depend on the re-transmission timer's time-out. The TCP Reno implements fast re-transmit and recovery algorithms. They also outperforms due to the burstiness of packet loss in a satellite environment. TCP New Reno fixes these problems with TCP Reno but is still not performing well⁸. One of the important concepts in TCP is window size. The maximum window size for TCP/IP is 65,535 bytes. The bandwidth-delay-product for satellite environment is very high. So the window size is not adequate to fill the channel with data. The window scaling option is also present which increases the window size to a maximum of 230 bytes (1GB) which is sufficient for Low Earth Orbiting (LEO) and Geostationary Earth Orbiting (GEO) systems. The satellite environment probably requires a window size equal to the bandwidth-delay-product for proper data transmission.

Path Maximum Transmission Unit (PMTU) discovery as defined in⁹ is performed to reduce the cost of fragmentation and reassembly. The PMTU size minus 40 bytes gives the maximum segment size. The Path MTU option helps to identify the most efficient packet size, but it causes some delay before TCP starts sending data due to the iteration of reducing an initial transfer unit 10, 11. The retransmission timeout value should be chosen carefully; otherwise unnecessary retransmits may lead to TCP performance degradation. Special consideration should be paid to the receiver sending delayed ACKs as explained in¹². In comparison with GEO satellite systems,

LEO constellations face delay-variations whose impact on TCP performance is currently a great issue¹⁰. The slow start phase unwisely spoils a lot of time during the time of connection establishment. If delayed ACK strategy is used then starting with an initial congestion window of one segment will result in a time-out at the receiver's side before sending the acknowledgement. RFC 2581¹¹ explains that a TCP can use an initial CWND of up to two segments. Further initial increase of the congestion window can still increase TCP's performance in the case of satellite links with high bandwidth-delay-product. The basic mechanism in TCP is the three-way handshaking. But in a satellite environment, the three way hand shaking adds an extra RTT to the latency. Especially for small and medium network, this causes extra cost and this in turn affects the performance efficiency¹⁴. Introduced a backward compatible option for TCP which would avoid this pitfall and is considered for implementation in satellite TCP stacks.

2.2 Network Policies

TCP performance can be improved by increased buffer size. A buffer size greater than 0.5 times the bandwidth-delay-product is sufficient to obtain a maximum throughput of 98%⁸. In Satellite networks with long propagation delay, buffering drop policies have no significant effect in terms of fairness and efficiency for TCP connections¹⁵. When RTTs reach values almost comparable to WAN latencies, the TCP performance gets improved. Distributing a minimum rate to all TCP connections equally over all link in high priority traffic increases TCP's efficiency by 30%. Rate guarantees provide performance improvement for both GEO and LEO satellite networks⁸. On comparing rate guarantees with end-system policies, we can say that rate guarantees do not increase the TCP performance much. But they should be considered as they assure a minimum flow of status information¹⁶. Introduces a Packet Pair algorithm for the throughput calculation that is performed over all IP traffics. Several other methods are also available to overcome the problems associated with satellite network. In some cases, certain flow control mechanisms are implemented at certain points in the path between the TCP end points^{17,18}. Satellite transport protocol can be used either in Split TCP connection or can be used as a transport layer protocol.

3. Transmission Control Protocol - An Overview

Transmission Control Protocol or TCP is an end-to-end connection oriented OSI transport layer protocol. TCP is a connection-oriented end-to-end highly reliable host-to-host protocol. They are used in packet switched computer communication network and in interconnected systems. It provides reliable byte-stream delivery service, independent of the underlying network architecture. It provides point-to-point full duplex data transfer. The TCP segment header fields holds the information agreed upon by both sender and receiver for synchronization and connection establishment through the network.

3.1 TCP Flow Control

TCP flow control means a fast sender does not overwhelm a slow Receiver. Flow control defines the amount of data a source can send before receiving an acknowledgement from receiver. The flow control must not be too slow. It must make sure that receiver does not get overwhelmed with data. TCP uses a sliding window protocol to accomplish flow control. For each TCP connection (always duplex), the sending and receiving TCP peer use this window to control the flow. The TCP is a reliable protocol for the data transmission. We can see problems only in the case of satellite communication. We can see large amount of data loss in the case of satellite communication, so an efficient TCP flow control mechanism is required. The existing flow control mechanisms include:

- Slow Start mechanism
- Congestion Avoidance mechanism
- Fast Retransmit
- Fast Recovery

Even after all these existing mechanisms, TCP is facing problems. The demand for satellite communication services is increasing. The TCP/IP is the most widely used data transmission protocol for communication. The characteristics of underlying physical layer have direct impact on performance of the TCP. So an efficient TCP flow control mechanism is required.

3.1.1 Slow Start and Congestion Avoidance

The network capacity can be measured by using a congestion window. At the time of establishing a TCP connection, the congestion window is set to one TCP segment. It increases by one segment on receiving each new acknowledgment. This window size increases until it reaches its maximum value of 64Kbytes. The sender maintains a retransmission timeout for the last unacknowledged packet. This retransmission timeout helps to detect the congestion. When the timer expires, the slow start threshold values get saved by the sender and set it to one segment. The sender then retransmits segments starting from the lost segment. The congestion window is increased by one segment on receiving each new acknowledgement and it increases until it reaches the previously saved slow start threshold. This state variable, the Slow Start Threshold (SSTHRESH), is used to identify whether the slow start algorithm or congestion avoidance algorithm is used to control data transmission. This is the slow start phase of TCP. After this, the congestion window increases by one segment during every RTT. This results in a linear increase of the congestion window for every RTT. This phase is called the congestion avoidance phase. If the delayed acknowledgement mechanism is implemented, then the duration of slow start become even longer as compared to the duration for normal slow start mechanism. If a TCP connection is in the congestion avoidance phase, the connection will not use it for a long time. As a result of this, the congestion avoidance phase in satellite networks with high RTT performs lower than that in a terrestrial network. The slow start and congestion avoidance can be described as follows. During the slow start phase of TCP, when a connection is established, the value of Congestion Window (CWND) is first set to 1 and after receiving ACK the value is updated to $CWND = CWND + 1$, which means the doubling of CWND for each RTT. The exponential growth of CWND continues until a packet loss is observed, causing the value of SSTHRESH to be updated to, $SSTHRESH = CWND/2$. After the packet loss is detected, the connection starts from slow start again with $CWND = 1$ and the window is increased exponentially until it equals the value of SSTHRESH. At this point, the connection goes to congestion avoidance phase where the value of CWND is increased to, $CWND = CWND + 1/CWND$, which implies the linear growth instead of exponential growth. This linear increase will continue until a packet loss is detected in the network.

3.1.2 Fast Retransmit and Fast Recovery

The TCP implementations use a timer for the retransmission timeout. This makes the TCP connection to wait a long time for the timeout. During this time phase, TCP will not send any new packets or retransmits lost packets. Moreover, once the timeout occurs, the congestion window is set to one segment, and the connection takes several roundtrip times to utilize the network efficiently. The fast retransmit and recovery algorithms are implemented by TCP Reno. If the network drops a segment, the subsequent segments arriving at the receiver can be out-of order. For each of them, the TCP receiver immediately sends an acknowledgement to the sender about this missing data segment. This acknowledgement is known as duplicate acknowledgement. When the sender receives three duplicate acknowledgements, it concludes that the segment referred by the acknowledgements has been lost and immediately retransmits the lost segment to the receiver. The congestion window is reduced by the sender. Whenever the sender receives an acknowledgement, the sender increases the congestion window size by one and tries to send a new data segment. Before sending one segment, the sender waits for half a round trip time, for each duplicate acknowledgement it receives. As a result of this, during the time of fast retransmit, the sender maintains the network link exactly at half capacity. At the same time, the TCP directly sets the size of congestion window to the slow start threshold. This is the fast recovery algorithm. Fast retransmit and recovery mechanisms are affected by long Round Trip Time over satellite links. Duplicate ACKs can be considered as one way of detecting lost packets. When receiving one duplicate ACK, the sender cannot understand whether the packet has been lost or just got out of order but after receiving several duplicate ACKs, it is possible to ensure that a packet loss has occurred. This makes the entire retransmission process very fast. This means that instead of waiting for the retransmit timer to expire, the sender can retransmit a packet immediately after receiving three duplicate ACKs.

The TCP connection always goes to slow start after a packet loss in the case of Tahoe. However, if the window size is large and packet losses are rare, it would be better for the connection to continue from the congestion avoidance phase, since it will take a while to increase the window size from 1 to SSTHRESH. The purpose of the fast recovery algorithm in Reno TCP is to achieve this behaviour. In a connection with fast retransmit, the

source can use the flow of duplicate ACKs to clock the transmission of packets. When a possibly lost packet is retransmitted, the values of Ssthresh and cwnd will be set to $Ssthresh = cwnd/2$ and $cwnd = Ssthresh$ which means that the connection will continue from the congestion avoidance phase and increases its window size linearly.

4. Factors Affecting TCP in Satellite Communication

A satellite communication system can be defined as a self-contained communications system. Satellites have the ability to receive signals from Earth and to retransmit those signals back with the use of a transponder. A transponder is an integrated receiver and transmitter of radio signals. A satellite has to withstand the shock of a launch into orbit at 28,100 km per hour. In addition, satellites have to be light, as the cost of launching a satellite is quite expensive. The transport layer protocol TCP plays an important role in data transmission in satellite communication. As the technology advances, the demand for satellite communication is increasing. TCP/IP is the most widely used communication protocol in satellite communication but it performs very poor. The major reason for this is the data loss due to congestion. The characteristics of satellite channel responsible for TCP/IP performance degradation includes the large end-to-end delay and delay variations. Satellite links with higher bandwidth causing a large bandwidth-delay product and a high Bit Error Rate (BER) as compared to the wired links as well as the burstiness of bit errors and the bandwidth asymmetry between the forward and reverse satellite channel.

TCP is a powerful transport layer protocol. On analyzing the TCP thoroughly, both theoretically and experimentally, it was found that a lot of factors affect the performance of TCP in satellite network. The TCP performance can be enhanced in three ways. They include the use of large window size, increasing the segment size and large initial window size. TCP limits the advertised receiving windows size to 65535 bytes. So by considering a large window size will enhance the TCP performances to a great extent. The satellite channel provides higher bandwidth but due to the poor performance of TCP, the available bandwidth is not utilized efficiently. As a result of this to enhance the TCP performance, a much larger segment size must be used. Satellite communication is

used mainly in highly critical interactive data transmissions. So the data transmission speed should be very high. This can be achieved by providing large initial window size. One downside of increasing the TCP window size is that it requires more memory for buffering on the server because all outstanding unacknowledged data must be held in memory. Another potential pitfall is in the case of performance when there is a packet loss because any loss packet within a window requires that the entire window be retransmitted. TCP is a self-regulatory protocol as the sender side enforces congestion control and receiver side enforces flow control. TCP performs the flow control by regulating the traffic speed so that the end host can process the data much faster. TCP controls the network congestion by regulating the speed such that the network can support traffic flow in such a way that the sender will not overwhelm the receiver. As discussed earlier, the slow start and congestion avoidance are the two important concepts coming under TCP. Both slow start and congestion avoidance strategy poses a lot of drawbacks. The slow start prevents new connections from flooding an already congested network. It forces an excessively long ramp-up for every new connection in satellite network. TCP assumes that all data loss occurs due to congestion. As a result, it responds by reducing the transmission rate. In satellite links, the RTT is very high. For example, the geosynchronous satellites orbiting at 22300 miles have a RTT of 540ms. The TCP misinterprets this long round trip time and bit errors as congestion and responds inappropriately. Moreover the maximum window size available is 64 Kb. But most of the OSs use default window size of 8KB. This imposes a maximum throughput rate of only 128 kbps per connection regardless of the higher bandwidth available. So the available bandwidth is not utilized efficiently. All these make TCP ill-suited for single-path satellite links. Beyond this the three major factors affecting the TCP performance in satellite network includes long propagation delay, bandwidth asymmetry and higher bit error rates.

4.1 End-to-end Delay and Delay Variation

The end-to-end delay includes the transmission delay and propagation delay. The propagation delay in communication system refers to the time lag between the departure of signals from the source and the arrival of the signal at the destination. This can vary from a few nanoseconds or microseconds in local area networks up to 0.25s

in geostationary-satellite communication systems. The total end-to-end delay in a satellite network can be considered as the sum of transmission delay, up-link delay and down-link delay, inter-satellite link delay, switching delay and buffering delay. In satellite network, the switching and buffering delay can be considered as negligible. The transmission delay can be calculated by dividing the total packet size by data rate. So the transmission delay is influenced by the data source itself due to the adaptation of packet sizes. The up-link delay can be obtained by dividing the total distance between the source and satellite by the total signal speed. In the same way the down-link delay can be calculated by dividing the total distance between the satellite and destination by the total signal speed. In the case of geostationary satellites having an orbital altitude of approximately 36,000 km, the delay for stations near the equator is around 120 ms. In the case of LEO satellites having a minimum elevation of 200 in between the sender and satellite and an orbital altitude of 1300 km, the minimum delay can be around 4.3 ms and the maximum delay becomes 12.7 ms. On considering the Inter-Satellite Link (ISL) distance, we can identify the difference very easily. The ISL distance for GEO satellite varies according to the number of satellites in the constellation. For example, in a constellation with 3 satellites, the ISL distance is about 73000 km. The ISL distance in the case of 12 satellite constellation becomes 21000 km. In the same way we can see the ISL distance variations in the case of LOE satellites also very easily. The entire throughput of TCP is restricted by the overall RTT of the satellite link. For a link with maximum sending window size of 64 Kbytes and RTT of 550ms, the TCP throughput is only 1Mb/second.

4.2 Band-width Delay Product (BDP)

The product of bandwidth and RTT gives the total amount of data in transit on a link at a given time. The satellite links are having very high latency and as a result of this, the BDP becomes very large. TCP/IP permits the remote host to send some amount of data without acknowledgment, which is important to achieve a good transfer rate in the case of connections with large bandwidth delay product. This data can be called as the TCP window size. The maximum window size is 64KB. In the case of satellite networks; the bandwidth-delay product plays an important role. If the window size is equal to the bandwidth-delay product, then the available bandwidth

can be utilized fully. The maximum throughput can be calculated by dividing the total window size by RTT. For example if the window size is 64KB then the throughput can be 64 KB/520ms. This increases the maximum data rate in the satellite network. The advertised window in the TCP header shows the amount of data that the receiver can receive i.e., it shows the receiver's available buffer size. For the purpose of efficient data transmission, the sender is not allowed to send more data than the advertised window. To increase the TCP performance, the sender and receiver set the buffer size not less than the Bandwidth delay product. To maximize performance, the sender should set its send buffer size and the receiver should set its receive buffer size to no less than the bandwidth-delay product. This buffer size has a maximum value of 64KB in most modern TCP/IP implementations. TCP acknowledgment spoofing is an important technique used for enhancing TCP performance. The bandwidth-delay-product is a major factor that affects the TCP performance when it comes to the concept of buffers or window sizes of the end system.

4.3 Bit Error Rates

BER (Bit Error Rate) is an important performance disrupting factor in GEO channel environment. TCP considers that the network is in congestion state when BER occurs in channel. So it reduces the efficiency of TCP transmission and also wastes available bandwidth in satellite channel. When the satellite signal is obscured, unexpected Bit Errors can occur. All these factors make TCP's performance unstable.

4.4 Bandwidth Asymmetry

The networking technologies exhibit asymmetry in their network characteristics that means the network characteristics in one direction may be entirely different than those in the opposite direction. In the case of satellite communications, the upstream bandwidth can be often limited compared to its downstream bandwidth. In some other cases, upstream communication on the same technology may simply be impossible. But this can be achieved by the use of different network technologies, like dialup modem line, etc., for upstream connectivity. Network asymmetry can cause adverse impact on the performance of feedback-based transport protocols like TCP. The network path can be both congested and uncongested. Even though the

network path in direction of data flow is uncongested, congestion in the opposite direction can cause disruption to the flow of feedback messages. This disruption can lead to poor performance. This performance degradation leads to the poor performance of TCP. Here in TCP, the feedback messages are in the form of acknowledgement packets. A disruption of the ACK stream can disrupt window growth and degrade performance to a fraction of the available bandwidth. A better protocol implementation for satellite network considers all the above issues into account in maximizing the throughput and the end user experiences. So by considering all the above factors we are proposing a new flow control mechanism for satellite network named Satellite-TCP.

5. Proposed Novel Flow Control Algorithm

The proposed new flow control algorithm is named as Satellite-TCP. Consider a satellite network topology with two gateways at each end of a bidirectional satellite link. Every TCP connection traversing the link is split

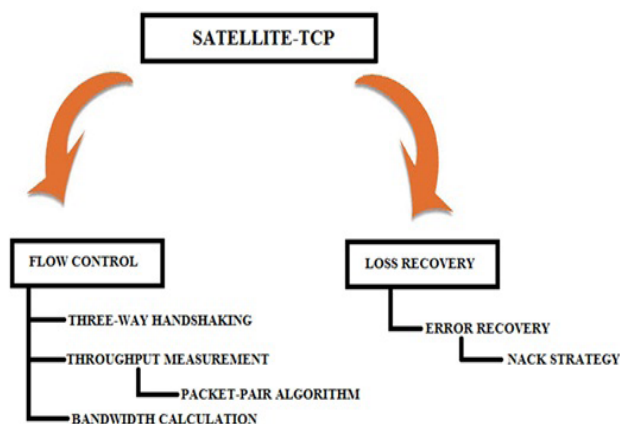


Figure 1. The Satellite-TCP architecture.

into three connections. One connection is established between the TCP sender and the Uplink Gateway (UG). The second connection is between the two gateways. The third connection is between the Downlink Gateway (DG) and the TCP receiver. The new proposed system or the Satellite-TCP improves the performance by using a new flow control and error recovery strategy. The entire system can be shown as follows:

Figure 1 shows the entire architecture of the new satellite flow control algorithm. The entire system is divided into two sections. One is the modified flow control strategy and the second part is the loss recovery strategy. The modified flow control works on the basis of the basic three-way hand shaking. The packet pair algorithm plays an important role in measuring the link throughput and this in turn helps to measure the available bandwidth in the satellite channel. The second section called the loss recovery is an error recovery mechanism. It considers the negative acknowledgements coming from the receiver side as the basic parameter for the error recovery.

5.1 Satellite-TCP Flow Control Mechanism

The three-way handshaking is the method used by TCP to set up a network connection. It is often referred to SYN-SYN-ACK as there are three messages transmitted for the purpose of connection establishment. Satellite-TCP makes use of this three-way handshaking mechanism. The sender does not perform any congestion window calculations. It sets the congestion window size to half of the receiving window size, i.e., $CWND = RWND/2$. Here CWND represents the congestion window and RWND represents the receiving window. The sender sends the data with a time interval, which is equal to $2 \times RTT / CWND$. The RWND is the window value advertised by the receiver. The window size is not measured in terms of bytes. They are measured in terms of MSS-sized segments. During the time of connection establishment, the receiving window size is adjusted as 1. On receiving the synchronous acknowledgement signal, the receiving window is set to the window field in the TCP header. The window value is calculated using a link utilization measurement mechanism. After this, the available buffer space for incoming data is calculated. The minimum of above two values is considered as the receiving window size. This is calculated by the downlink gateway. By using the pre-defined satellite link capacity, the link utilization can be calculated. For this the incoming throughput is also considered. We used the pre-existing packet pair algorithm for the throughput calculation. This packet pair algorithm is specified in¹⁶. The active and inactive links are distinguished using an idle timer. When this timer expires, the throughput calculation is set to zero. The next step is the available bandwidth calculation. First the total satellite link bandwidth is considered. Then the measured throughput is subtracted from this total satellite link

bandwidth, thus calculates the available bandwidth. This available bandwidth is calculated as it helps to identify how much more data can be allocated by the link. After calculating the available bandwidth, it should be distributed among all the active connections. This distribution should be in a fair manner. A connection or link considered can be active or inactive. One link can be considered as active only if it has transmitted some data during the last throughput calculation session. The inactive connections are not allocated with the available bandwidth share. The distribution of this available bandwidth among the active connections is done based on the receiving window size. But the receiving window size should be limited to a particular value, otherwise it will become unmanageable. So we fixed a maximum value for receiving window in such a way that, its value should not exceed this maximum value. This maximum receiving window is calculated using the round trip propagation delay of each link and the corresponding maximum segment size of the link. Whenever the connection link becomes inactive, the receiving window size is set to one segment. By setting the congestion window size to half of the receiving window size, it helps to increase the size of congestion window with higher speed. This ensures that the link will not experience any congestion. Thus the flow of data gets controlled.

5.2 Loss Recovery

Loss recovery is an error recovery process. It is based on the negative acknowledgement strategy. It is clear that the Satellite-TCP flow control mechanism prevents the congestion occurrence in the network. So the only problem that the network experiences is the bit errors and the resulting data loss. In normal TCP, the congestion window size gets reduced when a data loss is detected in the network but in the case of Satellite-TCP, the congestion window size is not reduced when a loss is detected. Instead of this, the congestion window is increased by the number of duplicate acknowledgements received. Now consider the highest sequence number of the packet sent when the first duplicate acknowledgement comes. If this sequence numbered packet is acknowledged, then the system enters the error recovery mode. Negative acknowledgement strategy is used to inform the missing data segments. Negative acknowledgements are a type of duplicate acknowledgement. On receiving this negative acknowledgement, the sender retransmits the requested

data along with new data. But here also the problem is with the number of duplicate acknowledgements. So to avoid this, a timer is used. The sender responds to the negative acknowledgements only when the timer expires. Initially this timer is set a value equal to one less than the receiving window size. On receiving each duplicate acknowledgement, the timer is reduced by one. Once this timer expires, it is reset. The new flow control mechanism uses the basic TCP's retransmission timer also to adjust the receiving window size and the amount of duplicate acknowledgement. The entire system of error recovery works by using different counters and flag values. When the first duplicate acknowledgement comes, its sequence number is used as the highest acknowledgement. At this time, the counter that records the number of duplicate acknowledgement is set to one. The highest sequence number of the negative acknowledgement is also considered for the retransmission of data. The retransmission counter is set to a value, one less than the receiving window size. The missed data segments requested by the negative acknowledgement are transmitted along with a certain amount of new data and the duplicate acknowledgement counter is reduced by the number of retransmitted segments. On arrival of the further duplicate acknowledgements, the duplicate acknowledgement counter gets incremented by one but the retransmission counter is reduced by one. Always the highest sequence number of the negative acknowledgement number is updated as the negative acknowledgement value in the header and the requested segment is retransmitted. After every iteration, the retransmission timer gets reduced and at last it reaches zero. At this time, the new data segments get transmitted. When this data segments successfully reaches the destination, the positive acknowledgement is received and as a result, all the counters and other parameters are reset and the same cycle get repeated.

6. Experimental Analysis

The network topology considered for the new satellite-TCP flow control mechanism, is shown in Figure 2. In the figure, the dotted arrow shows the new flow control mechanism applied in the satellite network. We evaluate the performance of new satellite-TCP by simulating the satellite network using the NS2 simulation tool. For the purpose of analyzing the performance of the new flow control mechanism, the Satellite-TCP is compared

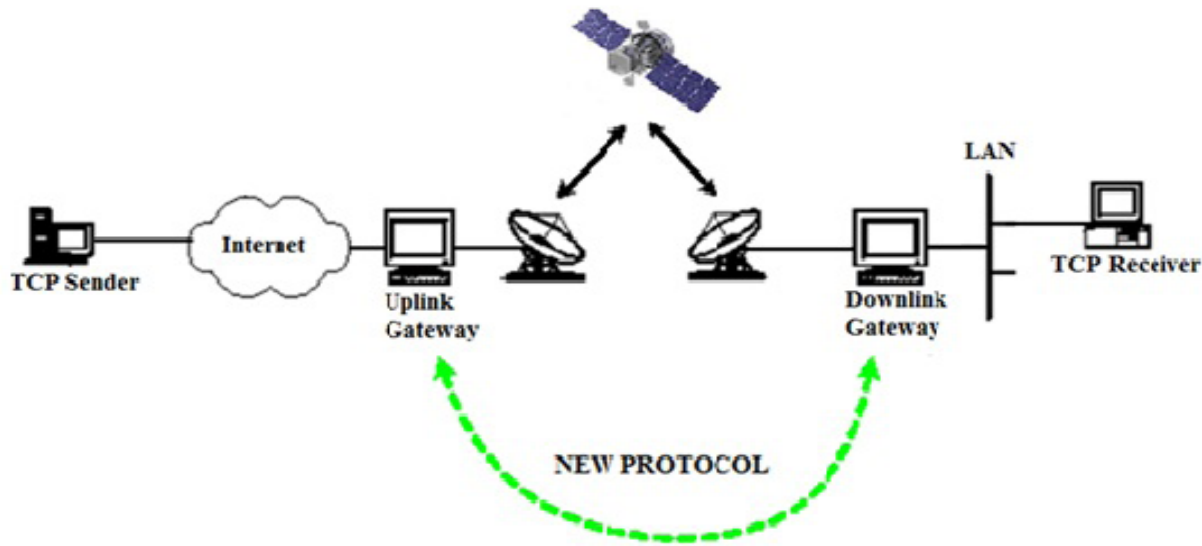


Figure 2. Topology for satellite network.

Table 1. Simulation Parameters

Parameters	Value
Packet Size	1500 bytes
Propagation Delay of satellite link	275 ms
RTT	550 ms
Link Capacity of satellite links	2-10 Mbps
Propagation Delay of non-satellite link	1 ms
Link Capacity of non-satellite links	10-100 Mbps
Receiving window value	64 packets
Queue Size	600 packets
Simulation Time	550 s

with TCP-New Reno and Peach. Here we are considering N number of senders and N number of receivers. These senders and receivers are connected to the uplink and downlink gateways respectively. These senders and receivers can perform data transmission of various file size. The parameters used for simulating the satellite network are shown in Table 1.

The satellite channels are studied in detail. The upper bound for TCP throughput is calculated by the equation:

$$\text{Max.Throughput} = \frac{\text{Receiving window size}}{\text{Round TripTime}} \quad (1)$$

Therefore, when using the maximum TCP receive window of 65,535 bytes over a satellite channel (RTT of 560 ms) the upper bound on throughput is given by equation:

$$\text{Max.Throughput} = \frac{65535 \text{ bytes}}{560 \text{ ms}} = 117,027 \text{ bytes per second} \quad (2)$$

This upper bound on TCP throughput ensures that TCP will be unable to fully utilize the available bandwidth provided by a satellite channel. To make the study more effective, simulation runs are carried out with different TCP variants and then the corresponding throughputs are calculated. Then the comparative study is carried out and in the same way the throughput calculations are carried out by applying the new flow control strategy also. Moreover the maximum window size required for the full utilization of available bandwidth can be calculated as follows:

$$\text{Window size} = \text{Bandwidth}/\text{RTT} \quad (3)$$

As shown in the Figure 2, a bi-directional GEO satellite link is used to establish communication between the data senders and the receivers. The data senders are connected to the Uplink Gateway (UG), and the receivers are connected to the Downlink Gateway (DG). They can transfer large amount of data of different file sizes. The data packet size is 1500bytes. The propagation delay of the satellite link is 275ms and that for non-satellite links is set to 1 ms. The RTT is 550ms. Link capacity values range from 2 to 10 Mbps in satellite links and for other links it ranges from 10 -100 Mbps. Queue sizes are set to 600 packets for all links. At first the entire experiment is carried out by taking single sender and single receiver. All other parameters are considered as explained earlier. For measuring the performance of new flow control mechanism, we measured the goodput. In computer networks, the goodput is the application level throughput. It is the number of useful information bits delivered by the network to the predefined destination in unit time. Here the overall overhead of the new mechanism and retransmitted data packets are excluded while calculating the goodput. For further experimentation, we calculated the overall throughput and compared it with that of other TCP variants like TCP-RENO, NEW RENO, Peach, etc. Here the most important parameter that should be considered is that the maximum available satellite link should be utilized. For this purpose, the receiver window size is varied from 100 to 500. The receiver window size is measured in terms of segments. We varied the values like packet loss probability value, file size and link capacity for the comparative study. The new Satellite-TCP performs better than the existing TCP variants as it is able to utilize the full available link in the satellite network. In normal cases, the congestion window size remains low due to

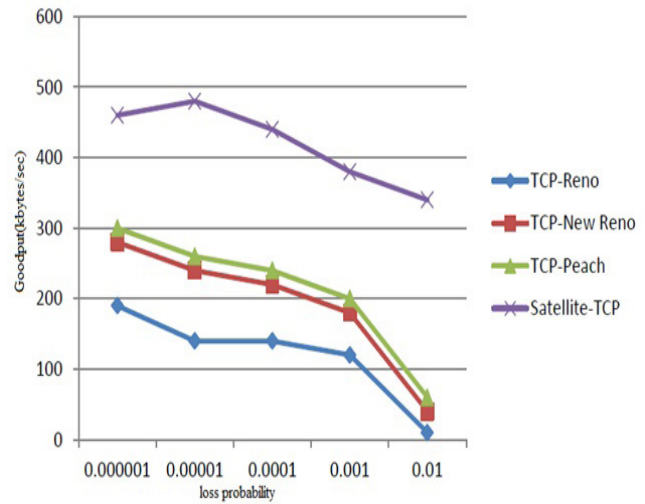


Figure 3. Good put values versus different packet loss probability values.

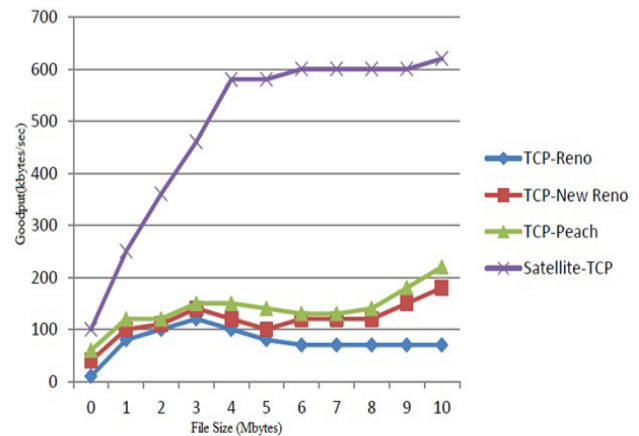


Figure 4. Good put values versus different file size.

data packet losses. But in the case of new Satellite-TCP, the congestion window size is raised to utilize the available link to its maximum. Figure 3 shows the Goodput achieved while running the simulation for Satellite-TCP. For plotting the goodput, we considered the packet loss probability. We set the file size as 1 Mbyte and the link capacity as 6Mbps. The performance of Satellite-TCP increases with file size and link capacities. Figure 4 shows the goodput achieved according to the varying file size and the Figure 5 shows the goodput values achieved for varying link capacity values.

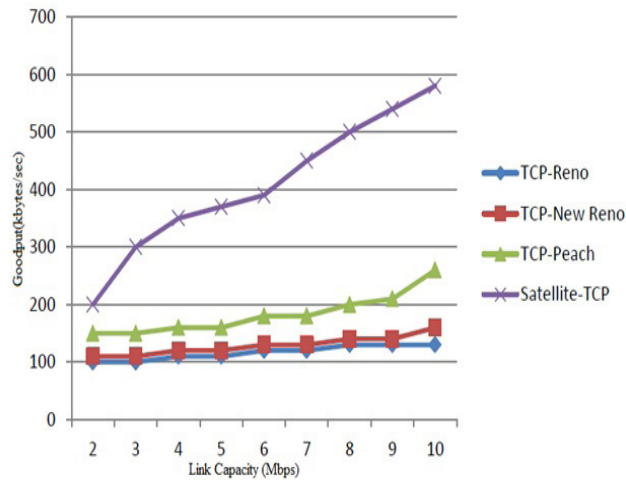


Figure 5. Good put values versus different link capacity values.

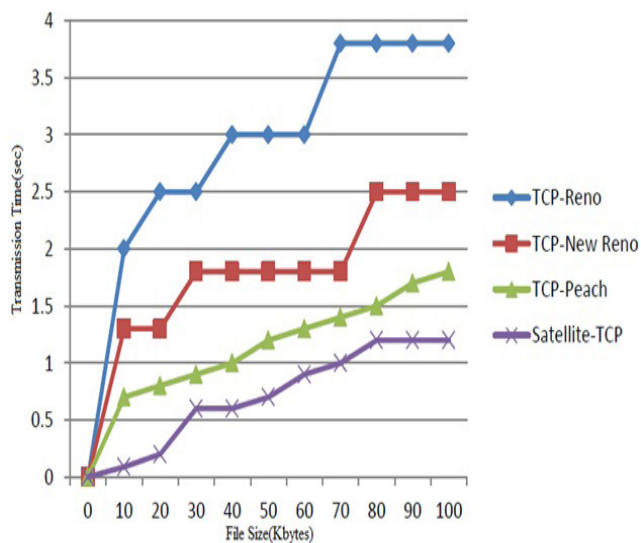


Figure 6. Transmission time versus the file size for single connection.

The performance can be measured in terms of transmission time also. In this case, Fig 6 shows that the Satellite-TCP's transmission time is shortest when file size is less than 64 Kbytes as compared to other variants of TCP like TCP-Reno, TCP-New Reno, and TCP-Peach. In the case of Satellite-TCP the congestion window get increased with a higher speed so that the files with smaller size get transmitted within a shorter time period.

The throughput calculated also showed the performance enhancement of Satellite-TCP as compared to

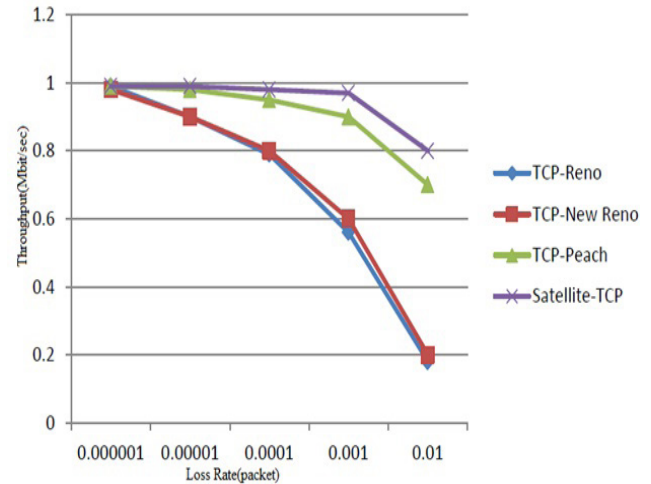


Figure 7. Throughput versus loss rate in the case of single connection.

others. As there is no scope for congestion in the network, we can say that the Bit error is the only reason for data loss. It is clear from the Figure 7 that the Satellite-TCP is having a higher throughput as compared to others. It is about 9.5 Mbits/second which is similar to others when packet loss rate is less than 10^{-4} . We can see from the graph that the throughput of other TCP variants decreases rapidly when loss rate increases. It is clear from the graph that the Satellite-TCP's throughput decreases slowly.

In the second phase of experiments, the number of senders and receivers, namely the number of connections is increased to 21. The problem in increasing the number of connection is that, there is a chance for congestion to occur in the network. This is because the maximum input rate of the network data is higher than the satellite forward link bandwidth. The maximum input rate can be calculated by the equation (4)

$$\text{Max. Input rate} = \text{Number of connections} * \text{Receiving window size} * 8 / \text{Round Trip Time} \quad (4)$$

The simulation is carried out in such a way that the data packets are sent in multiple time slots. The initial slow start phase is removed and the congestion window size is set as half of the receiving window size. This helps in full utilization of the link and also prevents the congestion to a great extent. The graphical analysis helped to prove that the new proposed flow control algorithm performs better than the existing TCP variants. It is clear that the Satellite-TCP distributes the link capacity in a

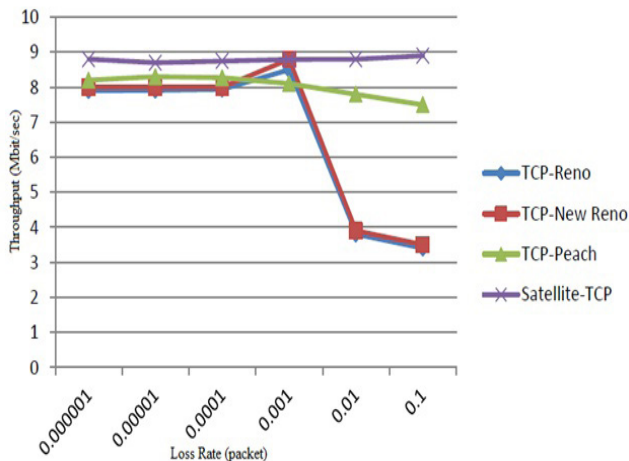


Figure 8. Throughput versus loss rate in the case of forward link.

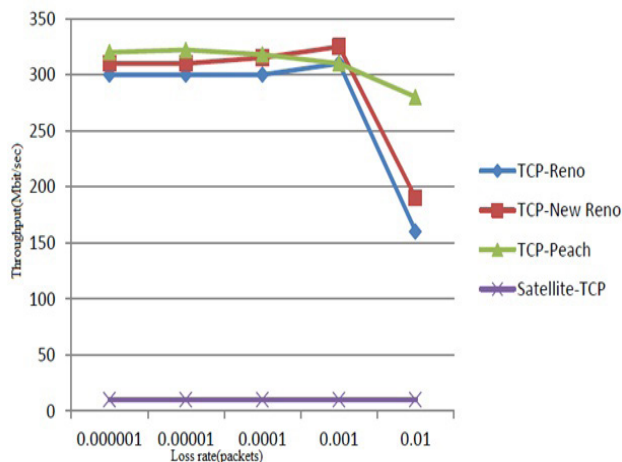


Figure 9. Throughput versus loss rate in the case of backward link.

better manner as compared with others. When the receiving window size is 64, RTT is 550 ms and the number of connection is 10, the forward link can reach its saturation level. Now we changed the number of connections to 21 and the backward bandwidth as 10 Mbits/s then the throughput variation is shown in Figure 8.

When the connection number is 21, the situation of backward bandwidth occupied is shown in Figure 9. The backward bandwidth of TCP variants continuously decline with increasing packet loss rate. But the Satellite-TCP's backward bandwidth is not affected by the packet loss rate as it is based on the NACK strategy, which is a

periodic response sending strategy. The backward bandwidth of Satellite-TCP occupies very low as compared to others.

For further clarification, the performance of Satellite-TCP, network with 21 connections are set with a link capacity of 6 Mbps. At first the sender begins transmission of a 2Mbyte file each. After a particular time period, sender sends a file of higher size. The receiver window size is adjusted in such a way that it will utilize the link without causing much congestion. The congestion window size is set to half of the receiving window size. The corresponding goodput is also calculated and it is clear from the experiments that, the Satellite-TCP distributes the link capacity in a fair manner as compared to other TCP variants.

7. Conclusion

In this paper we introduced a new flow control mechanism for satellite network, aiming at improving performance of satellite links. The proposed algorithm adopts a window growth strategy to accelerate the speed of increase in the size of congestion window. The flow control mechanism not only enhanced the throughput of forward link but also greatly reduced the bandwidth occupancy rate in back-ward link in the satellite network. The two major functions incorporated in this new algorithm are flow control and data loss recovery. Among this the flow control is based on link utilization measurements and throughput calculation techniques. Data loss recovery strategy is based on Negative Acknowledgements coming from the receiver side. Simulation experiments are carried out using the network simulator tool. The simulation results showed significant performance enhancement for the new flow control mechanism as compared with existing protocols. The simulation runs are carried out under varying parameter values and it exhibited remarkable fairness and considerable performance enhancement.

As a part of future enhancement we are planning to add a priority based strategy to the new flow control mechanism so that it can judge the network performance according to the data loss. Moreover we are expecting that, if we add a priority bit based strategy to the new flow control mechanism, it will help to distinguish the specific reasons for data loss and can take corresponding flow control strategies.

8. Acknowledgement

We offer our humble gratitude at the lotus feet of Sri Satguru Mata Amritanandamayi Amma, our spiritual mother who is the guiding light in our lives. We make use of this opportunity to express our sincere gratitude to all those who have helped us to complete this work successfully. We would like to express our sincere gratitude and thanks to the entire NISD team at ISRO, Thiruvananthapuram, for their continuous encouragement and supervision for the completion of this work. We wish to place on record our deep sense of gratitude to Dr. Latha Parameswaran, PhD, Chairperson, Dept. of Computer Science and Engineering, Amrita Vishwa Vidyapeetham, for her valuable guidance, help and useful suggestions. We express our sincere gratitude and thanks to Dr. Vidya Balasubramanian, PhD, Assoc. Professor, Dept. of Computer Science and Engineering, Amrita Vishwa Vidyapeetham, for her stimulating guidance, continuous encouragement and supervision throughout the course of present work. We also would like to extend our thanks and gratitude to Mr.A.K.Sumesh and Mr.M.Rithwik, Amrita Vishwa Vidyapeetham for their insightful comments and constructive suggestions to improve the quality of this research work. Last but not the least; we would like to express our sincere thanks to our friends for their valuable suggestions and helpful discussions.

9. References

1. Lin D, Morris R. Dynamics of random early detection. Proceedings of SIGCOMM97 Division of Engineering and Applied Sciences. Harvard University: Cambridge. 1997 Oct; 27(4): 127-37.
2. Hiroyuki Ohsaki MM, Miyahara H. Application of control theory to a window-based flow control mechanism with ECN routers. Osaka University: Graduate School of Engineering Science.
3. Liu Feng ZH, Hengna L. TCP-ATCA: Improved transmission control algorithm in satellite network. Chinese Journal of Aeronautics 21(2):155-61.
4. Wu J, Zhang P, Shi Y, Cheng S. ACK delay control for improving TCP throughput over satellite links. Proceedings of IEEE International Conference on Networks (ICON '99). 1999. p. 303-12.
5. Qu J. An enhanced TCP Vegas algorithm based on route surveillance and bandwidth estimation over geo satellite networks. 2010 IEEE International Conference on Measuring Technology and Mechatronics Automation (ICMTMA) and APCC; Beijing, China: Beijing University of Aeronautics and Astronautics; 2012. p. 806-11.
6. Henderson TR, Katz RH. Transport Protocols for Internet Compatible Satellite Network. IEEE Journal on Selected Areas in Communications. 1999 Feb 17: 326-44.
7. Goyal R, Jain R, Kalyanaramon S, Fahmy S, Vandalore B, Cai X. ATM Forum, Technical Working Group Members(AF-TM). ATM Forum, ATM_Forum/97-0424. 1999.
8. Goyal R, Jain R, Fahmy S, Vandalore B, Goyal M. Improving the Performance of TCP/IP over Satellite-ATM Networks. Design Issues for Traffic Management for the ATM UBR + Service for TCP over Satellite Networks. In: Jain R, editor. NASA Scientific and Technical Information Program. Hanover: NASA Center for Aerospace Information; 1999.
9. Mogul J, Deering S. Path MTU Discovery. Internet Engineering Task Force, Network Working Group. IETF, RFC 1191. 1990.
10. Allman M, Glover D, Sanchez L. Enhancing TCP over Satellite Channels using Standard Mechanisms. Internet Engineering Task Force, Network Working Group. IETF, RFC 2488; 1999.
11. Stevens W. The Protocols. Massachusetts: Addison-Wesley, TCP/IP Illustrated; 1998. p. 1.
12. Jacobson V, Braden R, Borman D. TCP Extensions for High Performance. Internet Engineering Task Force, Network Working Group. IETF, RFC 1323; 1992.
13. Allman M, Paxson V, Stevens W. TCP Congestion Control. Internet Engineering Task Force, Network Working Group. IETF, RFC 2581; 1999.
14. Available from: <http://www.rfc-editor.org/rfc/rfc1644.txt>
15. Goyal M, Goyal R, Jain R, Vandalore B, Fahmy S. Performance Analysis of TCP Enhancements of WWW Traffic using UBR+ with Limited Buffers over Satellite Links. ATM Forum, Technical Committee – Traffic Management Working Group. ATM Forum, ATM_Forum/98-0876R1; 1998.
16. Keshav S. A Control-Theoretic Approach to Flow Control. Proceedings of the conference on Communications architecture & protocols (SIGCOMM '91); Zurich: Switzerland; 1991 Sep. p. 3-15.
17. Border J, Kojo M, Griner J, G. Montenegro, Shelby Z. Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations. RFC 3135; 2001 Jun.
18. Minei I, Cohen R. High-Speed Internet Access Through Unidirectional Geostationary Satellite Channels. IEEE JSAC. 1999 Feb; 17(2): 345-59.