Applied Mathematics & Information Sciences

Volume 10
Issue 5 *Sep. 2016*

Article 33

9-1-2016

# An Efficient Explicit Error Notification with Adaptive Packet Size Mechanism to Improve TCP Performance in LEO Network

J. Govindarajan
*Dept of Computer Science and Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, Amrita University, India*, j_govindarajan@cb.amrita.edu

G. Kousalya
*Coimbatore Institute of Technology, TamilNadu, India*, j_govindarajan@cb.amrita.edu

Recommended Citation
Govindarajan, J. and Kousalya, G. (2016) "An Efficient Explicit Error Notification with Adaptive Packet Size Mechanism to Improve TCP Performance in LEO Network," *Applied Mathematics & Information Sciences*: Vol. 10: Iss. 5, Article 33.
DOI: http://dx.doi.org/10.18576/amis/100533
Available at: https://digitalcommons.aaru.edu.jo/amis/vol10/iss5/33

Applied Mathematics & Information Sciences
*An International Journal*

http://dx.doi.org/10.18576/amis/100533

# An Efficient Explicit Error Notification with Adaptive Packet Size Mechanism to Improve TCP Performance in LEO Network

*J. Govindarajan*[1,*] *and G. Kousalya*[2]

[1] Dept of Computer Science and Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, Amrita University, India
[2] Coimbatore Institute of Technology, TamilNadu, India

Received: 29 May 2014, Revised: 19 Dec. 2015, Accepted: 6th Jan. 2016
Published online: 1 Sep. 2016

**Abstract:** LEO satellite network plays a major role in the design of next generation internet. Due to high error rate of satellite link and absence of error handling mechanism in TCP, the end-to-end performance of existing TCP based applications over the satellite environment degrades substantially. Many research contributions have shown that with the help of explicit loss notification, sender is able to discriminate between loss due to congestion and loss due packet corruption, thereby avoids the unnecessary reduction of sending rate. Few studies, however, have mentioned that sending smaller size packets or optimum size packets can increase the success of packet delivery. The contribution of this work is to propose an integrated solution to improve the end-to-end performance of TCP using backward explicit notification of GSL (Ground-to-Satellite link) errors and optimum packet size calculation at TCP sender. With help of simulation experiments, we show that proposed scheme improves the end-to-end performance of TCP based applications over high error rate satellite links.

**Keywords:** Explicit error notification, optimum packet size, TCP, Satellite

## 1 Introduction

In Today's scenario, the diversified requirements of the Internet users are met through the interconnection of heterogeneous network segments (Optical network, Wireless network, Satellite network). For example, terrestrial network interconnected with satellite network to provide long distance communication, Internet facility to rural areas and backbone service between end user networks without wired communication infrastructure. However, high propagation delay and errors of satellite environment affect TCP (Transmission Control Protocol) mechanisms; thereby degrading the performance of existing TCP/IP based internet applications.

To provide the network stability, Jacobson & Michael (1988) [3] designed TCP with self clocking principle. In self clocking principle, sender's transmission rate is proportional to arrival rate of acknowledgement from the receiver. In wired network, arrival of acknowledgement is delayed due to congestion, whereas in satellite network data packet and acknowledgment packet deliveries are delayed by high propagation delay. Many research contributions have come to resolve this problem. TCP snooping and TCP splitting are the few solutions to handle this problem. Reducing the propagation delay by deploying of LEO (Low Earth Orbit) networks instead of GEO (Geostationary Earth Orbit) network, also avoids the delayed delivery and acknowledgment.

Wired communication infrastructure always exhibits high congestion. Nevertheless, reliable data transfer over wired environment was achieved by incorporating congestion handling mechanism in TCP. In contrast, satellite environment exhibits high bit error. If a node (Satellite router or receiver) detects error in the incoming packet through checksum validation, then the packet will be dropped. This packet drop createas a gap at TCP receiver or invokes timeout procedure at TCP sender. Gap at TCP receiver is reported to TCP sender using duplicate acknowledgment. In TCP, duplicate acknowledgement or timeout is an indication of packet loss due to congestion. Hence, CWND (Congestion Window) is reduced by half

[*] Corresponding author e-mail: j_govindarajan@cb.amrita.edu

or it reset to one and ssthresh (Slow-Start Threshold) is reduced to Flightsize/2. Flightsize is defined as amount of data that has been sent but not yet acknowledged [4]. In the case of Flightsize=1 (i.e., Sender has transmitted first packet after connection setup) and packet loss, ssthresh is set to two. This results in abrupt reduction of CWND. The absence of error handling mechanism in TCP invokes congestion control mechanism and reduces sending rate of TCP sender.

Many research contributions have come to find out the cause of packet loss for TCP over satellite network. In some of the contributions, a number of low priority data segments are sent along user data segments. Based on the number of acknowledgments received from the receiver, the packet loss classified either as congestion or link error [5]. Some contributions are based on flow splitting, where a customized transport is designed to handle the problems in satellite segment. However, these solutions are not measuring or reporting the errors explicitly. The classification of packet loss is based on prediction procedure or indirect observation.

In contrast to these contributions, explicit congestion notification [6] and explicit error notification schemes [7, 1,2] have been designed to enable the TCP sender to distinguish packet loss due to congestion from packet corruption. In [7], explicit error notification scheme has been discussed in detail. This paper proposed cumulative ETEN (Explicit Transport Error Notification) scheme, in which intermediate node estimates the survival probability of link and reported to the source using forward explicit notification. Using explicit report about packet loss, TCP sender is able to distinguish packet loss due to congestion from packet corruption. During the recovery of corrupted packet, packet is retransmitted and window size is maintained without reduction. However in the case of prolonged link error, the retransmitted packets will get corrupted and leads to further retransmission. Hence, error notification mechanism with retransmission could not improve TCP performance significantly.

At the same time, some research analyses have done by researchers to understand the relationship between packet size and BER (Bit Error Rate) [9,10]. In [10], relationship between BER and Packet size has been studied and optimal protocol packet size equation has been derived. Reducing the packet size during high error rate and increasing the packet size during low error rate will increase packet delivery efficiency (packet delivery efficiency is the ratio of usable packets to transmitted packets).

In summary, explicit error notification scheme is most suitable scheme for wireless and satellite network. To get the enhanced TCP performance over satellite environment, explicit notification must be supported by some mechanism to increase the probability of successful packet delivery. The aim of our work is therefore to propose a backward explicit error notification scheme with optimum size packet to improve the performance of TCP end points.

The rest of this paper is organized as follows. In section 2, we present our simulation results to understand the relationship between BER and packet size. This is followed by a summary of works related to handling bit errors in section 3. In section 4, we describe our proposed scheme. In section 5, we evaluate the proposed scheme for TCP over LEO satellite network through experiments. Finally, we conclude with some directions toward future works in section 6.

## 2 RELATIONSHIP BETWEEN BER AND PACKET SIZE

To understand the relationship between BER and Packet size in LEO satellite network, the following set of experiments are performed using ns-2 simulator. The aim of these experiments is to demonstrate that sending smaller packets over high error rate link will increase the probability of successful transmission.

### 2.1 Simulation environment

We use the topology shown in Figure 1 to conduct our experiments. The set of constellation parameters used to model the LEO network is shown in Table 1. One pair of Sender and Receiver is considered. TCP Newreno and FTP (File Transfer Protocol) are configured as transport protocol and application protocol respectively. A set of packet sizes (1500, 500, 200, 100 and 50 bytes) and a set of BER (0.01, 0.001 and 0.0001) are taken for the simulations. TCP efficiency and throughput are taken as performance metrics. TCP Efficiency [13] is defined as the percentage of transmitted bytes which are successfully transmitted without retransmission.

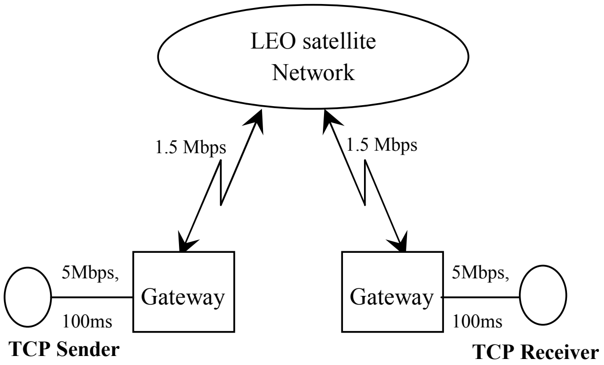

**Figure 1:** LEO scenario used for the experiments

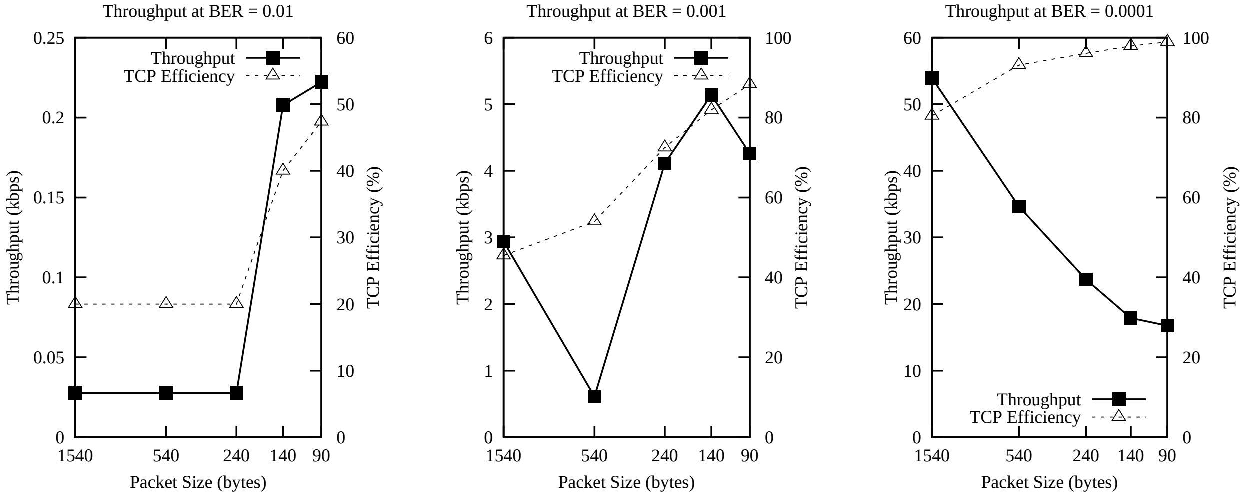

**Figure 2:** Performance of TCP Newreno for various BER and Packet size (bytes)

**Table 1:** Constellation Parameters

| | |
|---|---|
| Total Number of Satellites | 66 |
| Number of Planes | 6 |
| Number of Satellites Per Plane | 11 |
| Satellite Altitude | 780km |
| Eccentricity | 0.002 |
| Inclination Angle | 86.4° |
| Inter-satellite Separation | 360/11° |
| Inter-Plane Separation | 31.6° |
| seam separation | 22° |
| Minimum elevation angle at edge of coverage | 8.2° |

### *2.2 Simulation results*

The simulation results are presented in Figure 2. The results show that sending smaller size packets at high BER will result in increased TCP efficiency. For example, consider the case of $BER = 10^{-3}$. In this case, packet size of 1540 bytes gives 50% of TCP efficiency and packet size of 140 bytes gives 80% of TCP efficiency. If one packet with size of 1540 bytes (including 40bytes of header) corrupted, then it is a loss for user data of size 1500 bytes and it causes for retransmission for 1500 bytes of user data. Similarly, packet corruption of one packet with size of 140 bytes (including 40bytes of header) causes retransmission of 100 bytes. Even though packet corruption invokes the retransmission in both cases, the amount of retransmission is less in the case of smaller size packet compared to the case of larger size packet. The reduction in amount of retransmission maximizes TCP efficiency and improves the end-to-end performance.

## 3 HANDLING BIT ERRORS: RELATED WORKS

To provide next generation communication services using satellite network, an important design issue is to improve the protocol efficiency from loss due to bit corruption. The traditional solutions to handle the bit error problems are Forward Error Correction (FEC) [20,21] and Bit interleaving. In FEC, the receiver end point uses the redundant information attached by the sender to correct bit errors. Even though FEC scheme quickly recovers the packet using bit correction, it is not efficient solution for the links with high BER. Bit interleaving, reduces the likelihood of consecutive bit corruption by spreading the sequence of bits across multiple packets. But loss of one packet due to error will create multiple gaps at application level of the receiver. Corruption of single bit or multiple bits of a packet will result in dropping or rejection of all bits. Instead of injecting single large size packet into the link, transmission of multiple small size packets will increase the likelihood of successful transmission. At low BER, probability of bit error is significantly less and hence packet dropping probability is also less. At high BER, probability of bit error is significantly high and hence packet dropping probability is also high. Based on these observations, various studies were performed [9, 10]. Dynamic Packet size mechanism (DPSM) [14] has been proposed to send variable size packets between Mobile Host (MH) and Base Station (BS) using the feedback from BS. DPSM uses adaptive factor to increase and decrease the packet size. In [10], equation for optimum packet size (i.e., packet size which maximizes the information efficiency) has been derived.

$$S_p = \sqrt{\frac{S_h n}{BER}} \tag{1}$$

Where, Sp - Packet Size, Sh - Packet header size, n - Error density factor (n=1 for the worst case)

## 4 PROPOSED SCHEME

We introduce the idea of extending TCPNewreno protocol, to enable a TCP sender to respond for Backward Explicit Transport error notification and BER estimation report from the gateway by retransmitting the notified packet without decreasing the congestion window and calculating optimum packet size respectively. For BER estimation at packet level, we use Maximum Likelihood (MLE) mechanism [8]. In this work, we consider the errors in GSL (Ground-to-satellite Link) and errors in Inter Satellite Link (ISL) are not considered. A router can use either forward notification or backward notification to report the error to the source of the packet. In forward error notification, router attaches the report with packets on the path to its destination and then destination communicates to source about the error by piggybacking the acknowledgment packet. In backward error notification, router communicates about the error to the source directly using explicit packet. Backward Explicit notification, in comparison to Forward Explicit Notification, takes very less time to communicate about the error to the source and source initiates the retransmission quickly. Hence, we design our solution using backward explicit notification.

The communication between a LEO satellite, a Gateway and an End node (TCP source) of our proposed scheme is summarized in Figure 3. Each LEO satellite verifies the incoming packets from the gateway for the error and communicates the error information to gateway. Using the feedback from the Satellite, gateway recalculates the link (Gateway - Satellite) quality in terms of BER and communicates the BER to ”end nodes” by piggybacking. Also it sends error notifications to the respective sources. The end nodes retransmit the reported packets and/or recalculate the optimum packet size. We have added an additional packet processing procedure and communication procedure in a satellite with the assumption that the recent satellites are having the Onboard Processing capability. In [18,19], Onboard Processing capabilities like onboard processing including modulation & demodulation, IP routing & caching and participation in QoS (Quality of Service) protocols have been summarized. Next, we discuss BER estimation, Backward Explicit notification, and procedure of optimum packet size calculation and retransmission procedure for TCP end points in the following subsections.

### *4.1 BER Estimation*

In our solution, using the received error information (packet is corrupted or not corrupted) from the LEO satellite, BER estimation is performed continuously by the gateway to assess the GSL. We have chosen Maximum Likelihood (MLE) [8] to estimate BER using only packet level information without using radio level information.

#### 4.1.1 BER Estimation using MLE [8]

Under the high BER, packet corruption probability of consecutive packets is high. The sequence of unsuccessful packet is called as gap. BER of sample packets is calculated using sample number of packets (n), total number of bits in the n packets (N), number of lost packets (m) and total number of bits in those lost packets (M).

$$\text{Bit Success Rate}(\alpha) = \left(1 - \frac{M}{N}\right)^{\frac{m}{M}} \tag{2}$$

$$\text{Bit Error Rate}(1-\alpha) = 1 - \left(1 - \frac{M}{N}\right)^{\frac{m}{M}} \tag{3}$$

#### 4.1.2 Enhancement to MLE

In the literature [8], BER estimation of sample packets is given with window size 10 to 10,000. Window size of 10000 will take very long time for the estimation. To calculate the BER quickly, we choose window size = 10. However, BER estimation with sample size of 10 gives temporary fluctuations. In order to remove the temporary fluctuations, we use Exponential Weighted Moving Average as smoothing method.

$$\text{Bit\_Sample} = 1 - \left(1 - \frac{M}{N}\right)^{\frac{m}{M}} \tag{4}$$

Exponential Weighted Moving Average

$$\text{BER\_estimated} =$$

$$(1-\beta) \times \text{BER\_estimated} + \beta \times \text{BER\_sample} \tag{5}$$

Here, $\beta$ is assumed as 0.2

### *4.2 Optimum packet size calculation using BER estimation:*

Equation (1) has been used to calculate the Optimum Packet Size (Sp). However, equation (1) yields fixed

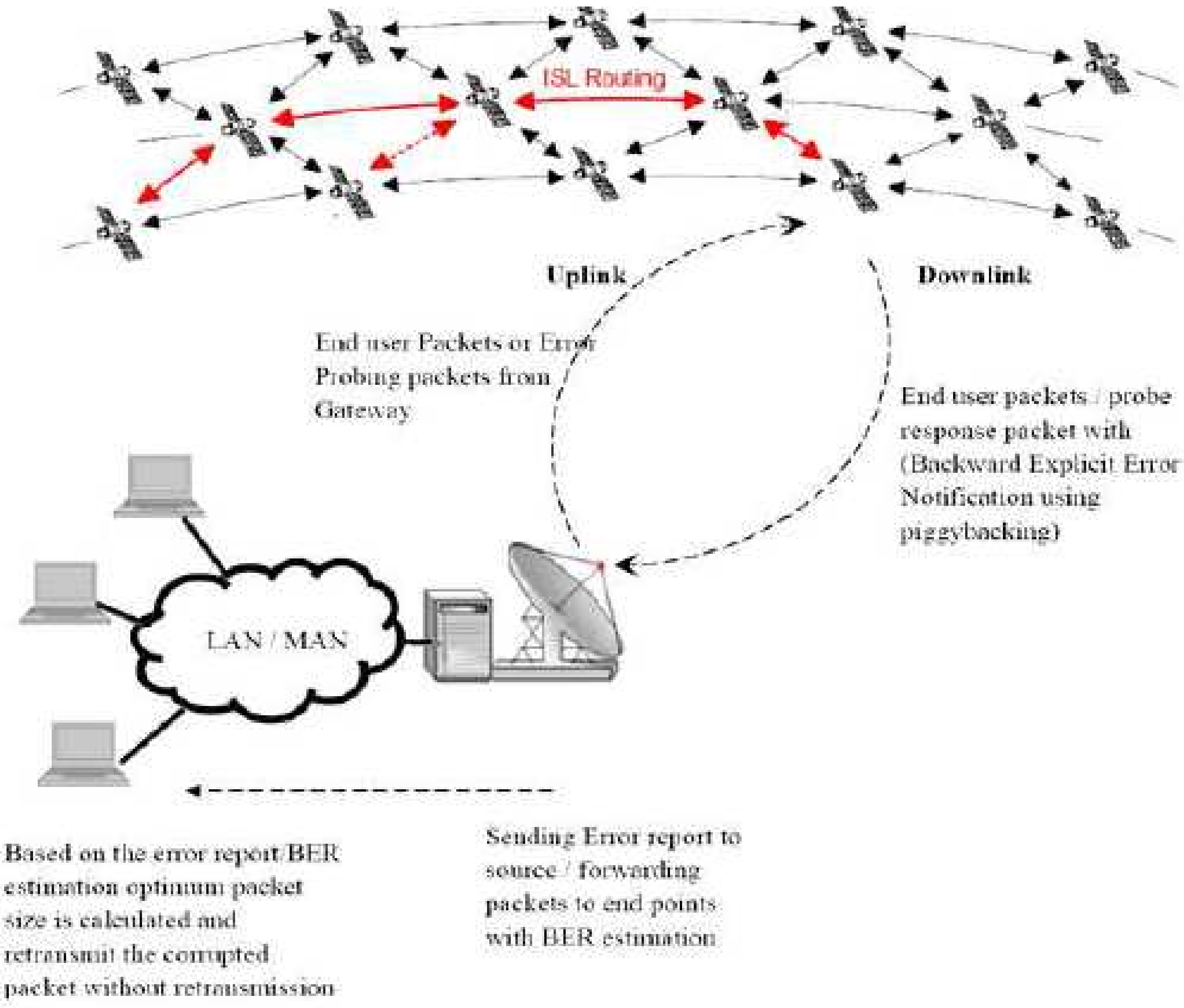
ISL Routing
Uplink
Downlink
End user Packets or Error Probing packets from Gateway
End user packets / probe response packet with (Backward Explicit Error Notification using piggybacking)
LAN / MAN
Based on the error report/BER estimation optimum packet size is calculated and retransmit the corrupted packet without retransmission
Sending Error report to source / forwarding packets to end points with BER estimation

**Figure 3:** Illustration of the proposed scheme

combination of BER and packet size. For example Sp=2236 bytes at $BER = 10^{-6}$ and Sp = 70 bytes at $BER = 10^{-3}$. Even though optimum packet size increases the success rate, sending smallest packets and largest packets will create new problems. Smallest size packets (for example, 20bytes) will increase the overhead and results in performance degradation. Largest size packets may exceed the MTU (Maximum Transmission Unit) of given path and results in fragmentation or packet drop. To provide some flexibility or to restrict the packet size for the specific implementation, the calculated optimum packet size limited to minimum and maximum packet size. If calculated optimum packet size is less than minimum packet size, then minimum packet size will be taken as optimum packet size. Similarly, if calculated optimum packet size is greater than maximum packet size, then maximum packet size will be taken as optimum packet size.

### 4.3 BEEN - Backward Explicit Error Notification

In case of receiving a packet from the gateway (uplink), checksum validation is performed by the satellite. If error is detected, then a record (Pktid, BEEN) is added to ErrornodeQueueList table and the packet is dropped. Here, Pktid indicates Packet Id and BEEN indicates error flag. If downlink queue is empty, a special packet (probe response packet) is generated and inserted to downlink queue. Before forwarding a packet from downlink queue, a record from ErrornodeQueueList table is removed and inserted into option field of IP header. Reactions of the

Gateway and TCP sender to the received packet are summarized in Algorithm 1 and Algorithm 2.

**Algorithm 1:** Gateway - BEEN - Backward Explicit Error Notification

```
initialization;
$M \leftarrow 0$ $N \leftarrow 0$ $m \leftarrow 0$ $n \leftarrow 0$
On forwarding each a packet to LEO Satellite (Uplink);
begin
    Set a Packet ID in option field of IP header
end
On Gateway receives an end user packet or a probe packet with PacketID and BEEN from LEO Satellite (Downlink);
begin
    $n \leftarrow n+1$ $N \leftarrow N + Packetsize$
    if $BEEN = 1$ then
        $m \leftarrow m+1$
        $M \leftarrow M + sizeofthepacket$
        send error report to source with (seqno, srcport, estimated BER)
    end
    if $n = 10$ then
        calculate BER using MLE
        $M \leftarrow 0$ $N \leftarrow 0$ $m \leftarrow 0$ $n \leftarrow 0$
    end
    set estimated BER in the packet
end
```

**Algorithm 2:** TCP Sender's reactions to the received packet

```
On Receiving error report (with estimated BER) about one of its previously transmitted packet ;
begin
    Calculate optimum packet size using eq. (1)
    Retransmit the corrupted packet without reducing the CWND
    if $seqno = highestack$ then
        Reset the retransmission timer
    end
end
```

## 5 SIMULATION RESULTS AND DISCUSSION

To study the end-to-end performance of our proposed scheme, SaTPEP (Satellite TCP Performance Enhancing Proxy) [11] which is based on TCP split connection and standard TCPNewreno [12] have been chosen. The simulations were performed using ns-2 Simulator to evaluate performance of our proposed scheme in comparison with TCPNeworeno and SaTPEP. The implementation of our scheme has been done by extending the standard implementation of TCPNewreno in ns2. For our proposed scheme, Gateway needs to communicate with LEO satellite and TCP sender. To provide communication with TCP senders, ErrorReporting Agent was designed for both Gateway and End node. ”ErrorReporting Agent” in gateway is responsible for generating and sending error report to ”Errorreporting Agent” of End nodes. Agent in the end node side receives the report which is sent by the gateway and forwards it to respective TCP sender. At end node, ”TCPNewreno” Agents are connected with ”Errorreporting” agent.

### 5.1 Simulation environment

The constellation parameters specified in section 2 are used to model the LEO network. FTP is configured as application protocol. A set of packet sizes (1500, 500, 200, 100 and 50 bytes), a set of BER (0.01, 0.001 and 0.0001) and transport protocols (Proposed scheme, TCPNewreno and SatPEP) are taken for the simulations. For each simulation, Transport protocol, packet size and BER are changed. Simulation time is configured as 200 sec. For our proposed scheme, minimum and maximum packet sizes are configured as 80 bytes (including 40bytes of header) and 1540 bytes (including 40bytes of header) respectively. In our simulations, the Maximum packet size is referred using the variable ”Initial packet size”. Since TCPNewreno and SaTPEP are based on fixed packet size, the variable ”Initial Packet size” indicates the size of the packet. For the proposed scheme which is based on adaptive packet size, this variable indicates maximum packet size. Hence, sender sends the first packet with the configured initial packet size and the subsequent packets will be sent with adaptive packet size. Header size of 40bytes is fixed for all simulations.

### 5.2 Performance metrics

To validate the performance of our proposed scheme, the performance metrics Throughput, Goodput, Total amount of user data transferred to the receiver, TCP efficiency, Information efficiency are considered. Throughput (kbps) represents average rate of transferred data to the receiver. It is the ratio between total data (bits) received by the receiver including retransmitted data (bits) and simulation time. Goodput (kbps) represents the average rate of successfully transferred data to the receiver. It is the ratio between successfully received data (bits) excluding retransmitted data (bits) and simulation time (sec). Also we compare the schemes interms of total amount of user

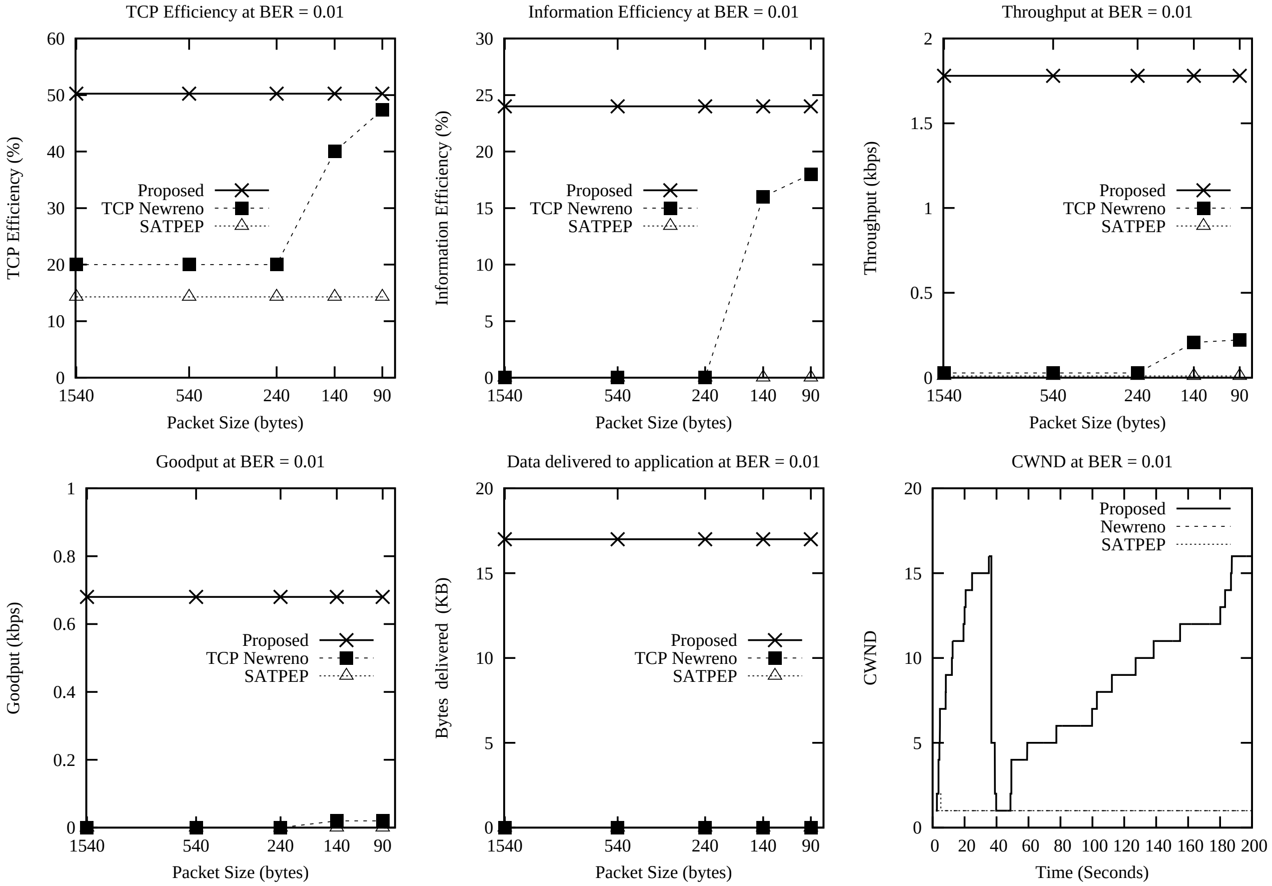

**Figure 4:** Performance of Proposed scheme, TCP Newreno and SatPEP at BER = 0.01

data delivered to the receiver. It is specified by bytes delivered (KB). TCP Efficiency [13] is defined as the percentage of transmitted bytes which are successfully transmitted without retransmission. i.e., ratio between non retransmitted bytes and total transmitted bytes. Transmitted Bytes are the total number of TCP Bytes to be transmitted, including the original and the retransmitted Bytes.

$$\mathrm{TCPEfficiency}(\%) = \frac{\mathrm{TransmittedBytes} - \mathrm{RetransmittedBytes}}{\mathrm{TransmittedBytes}} \times 100 \qquad (6)$$

The information efficiency [10] is calculated by multiplying packet delivery efficiency (ep) with packet header efficiency (eh). Packet header efficiency (eh) is a fraction of packet size allocated for end user data. Packet delivery efficiency (ep) is a ratio between the number of successfully transmitted packets and the total number of packets transmitted.

$$\begin{aligned}\mathrm{InformationEfficiency} &= \\ &\mathrm{PacketHeaderEfficiency}(eh) \\ &\times \mathrm{PacketDeliveryEfficiency}(ep) \qquad (7)\end{aligned}$$

### 5.3 Simulation results

To show the improved performance in the case of high BER, we conducted simulation experiments with BER of 0.01, 0.001 and 0.0001. The results of simulation experiments are summarized in the following subsections.

#### 5.3.1 Performance analysis with BER = 0.01

With BER of 0.01, the frequency of the packet corruption is very high. Figure 4 presents the results of simulations with Bit Error Rate of $10^{-2}$. As shown in the Figure 4, the performance of TCP Newreno and SatPEP are very poor compared to our proposed scheme. At BER of 0.01, optimum packet size using equation (1) is 22bytes including header. Since it is less than the configured minimum packet size (80 bytes) and configured maximum packet size (1540, 540, 240, 140 and 90 bytes), the proposed scheme has taken 80 bytes (i.e. 40 bytes of data with 40bytes of header) as optimum packet size for all initial packet sizes. Because of optimum packet size calculation and explicit error report notification, the proposed scheme achieves 50% of TCP efficiency and 24% of Information efficiency with all initial packet sizes. TCP Newreno, though achieves 50% of TCP efficiency and 18% of information efficiency with 50 bytes of packets, the achieved throughput and goodput, and amount of bytes delivered to the receiver are less compared to the proposed scheme. As per TCP Congestion control specification, initially sender enters into ”slow start” phase with CWND = 1. In case of timeout event, TCP sender resets the CWND to 1 and ssthresh to Flighsize/2. For the packet sizes greater than 200 bytes, we observed that first packet is dropped and timeout is invoked in TCP sender. Because of timeout procedure at CWND = 1, ssthresh became 2. This reduction in ssthresh affected the growth of CWND in TCPNewreno and SatPEP. In our proposed scheme, drop of first packet has been reported to the source explicitly in less time and retransmitted immediately. Hence, timeout event avoided and significant reduction of CWND due to timeout event in our scheme have been avoided. A significant difference in growth of CWND for the proposed scheme can be observed compared to other schems. Quick retransmission using the explicit report and optimum packet size calculation, avoids the CWND reduction and increase probability of success.

### 5.3.2 Performance analysis with BER = 0.001

Figure 5 presents the results of simulations with Bit Error Rate of $10^{-3}$. In this case also, we have observed the improved performance for our scheme compared to the performance of TCP Newreno and SatPEP. At BER of 0.001, optimum packet size is 70 bytes including header. Since it is less than the configured minimum packet size (80 bytes) and configured maximum packet size (1540, 540, 240, 140 and 90 bytes), the proposed scheme has taken 80 bytes (i.e. 40 bytes of data with 40bytes of header) as optimum packet size for all initial packet sizes. For the configured maximum packet size of 90 bytes, small decrease in the performance has been observed for our scheme. Nevertheless, it has achieved higher throughput, goodput and amount of bytes delivered compared to TCPNewreno and SatPEP. The proposed scheme achieves 90% of TCP efficiency and 57% with all initial packet sizes. TCP Newreno and SaTPEP though achieve 90% of TCP efficiency with 90 bytes of packets and 65% of information efficiency with 240 bytes, the achieved throughput and goodput, and amount of bytes delivered to the receiver are less compared to the proposed scheme.

Explicit error report and retransmission of our scheme avoids timeout, and maintains the sender in congestion avoidance and fast recovery state. Because of transmission during fast recovery phase, CWND of the sender inflates initially. Then there is a deflation in congestion window (CWND) followed by increase in congestion window (CWND). This deflation of CWND is to bring the fast recovery phase to an end [15]. Nevertheless, a significant growth of CWND can be observed compared to TCPNewreno and SatPEP.

### 5.3.3 Performance analysis with BER = 0.0001

Figure 6 presents the results of simulations with Bit Error Rate of $10^{-4}$. In this case also, we have observed the improved performance for our scheme compared to the performance of TCP Newreno and SatPEP. At BER of 0.0001, optimum size of a packet is 224 bytes. When initial packet size (240bytes, 140 bytes and 90 bytes) is less than optimum packet size (224 bytes), a decrease in throughput and goodput, and an increase in overhead percentage are observed for all three schemes. TCP efficiency achieved by the proposed scheme, TCP Newreno and SatPEP are closer. Nevertheless, the proposed scheme has higher throughput, goodput and amount of data delivered to the receiver. In the absence of error handling mechanism in TCPNewreno and SatPEP, packet drops have created significant reduction in CWND of the sender. In proposed scheme, CWND of the sender increases exponentially and reaches ssthresh. After reaching ssthresh, sender enters into congestion avoidance and increasing linearly.

## 6 CONCLUSION

In this work, we have proposed an efficient backward explicit error notification scheme integrated with optimum packet size calculation. Backward Error notification to the TCP sender can avoid the false reduction of TCP CWND (Congestion Window). Gap-MLE [8] has been used to estimate BER of Ground-satellite link at packet level. The gateway attaches BER report with all packets of downlink. Based on the BER report from the received packets, existing connections or new connections are able to calculate optimum packet size to reduce the error probability of future packets. We also have demonstrated that our solution will improve end-to-end performance of TCP. In LEO network, end nodes are connected by ISLs and GSLs. Our solution has been designed to report the packet drops due to GSL errors only. Hence, reporting mechanism for packet drops due to ISL errors needs to be studied further. In [10], relationship between optimum packet size and information efficiency has been studied in terms of both deterministic and probabilistic. In this work, optimum packet size calculation which optimizes

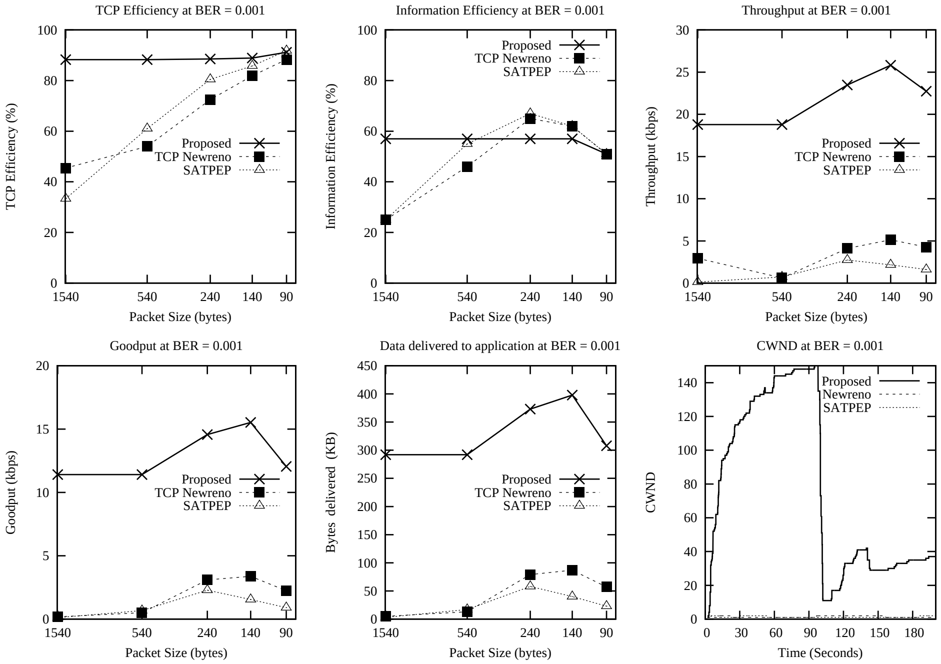

**Figure 5:** Performance of Proposed scheme, TCP Newreno and SatPEP at BER = 0.001

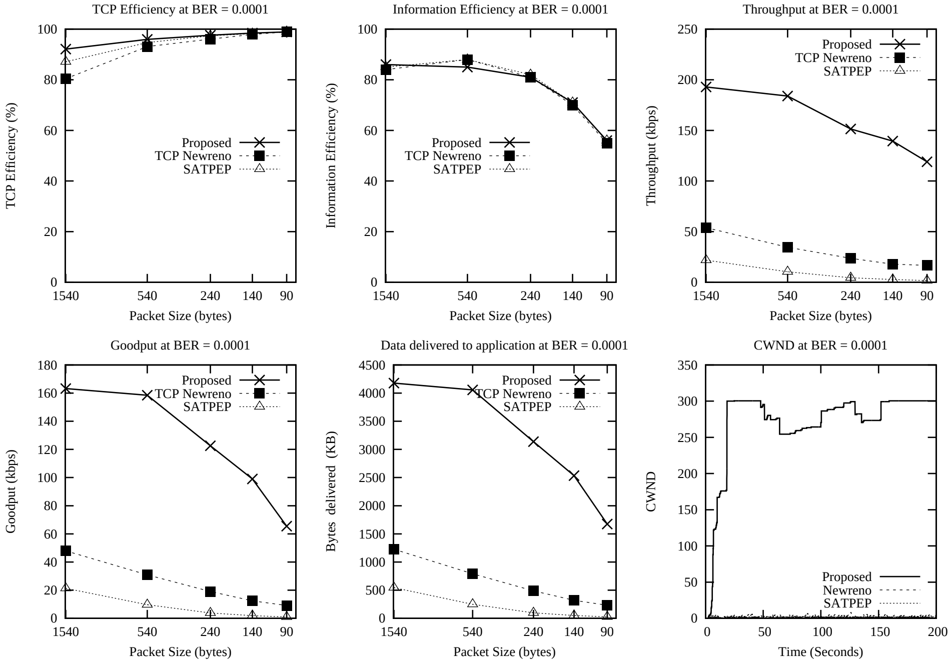

**Figure 6:** Performance of Proposed scheme, TCP Newreno and SatPEP at BER = 0.0001

the efficiency of deterministic packet delivery has been used. The Deterministic calculation can be replaced by the probabilistic calculation and its results can be further investigated.

## References

[1] Buchholcz, G., Gricser, A., Ziegler, T., Van Do, T.: Explicit Loss Notification to Improve TCP Performance over Wireless Networks. In: Freire, M.M., Lorenz, P., Lee,

M.M.-O. (eds.) HSNMC 2003, vol. 2720, pp. 481-492. Springer, Heidelberg (2003).

[2] Garcia, J. ”Explicit and Implicit Loss Notification for Error-prone Links”, Proceedings of Promote IT 2003, Visby, Sweden, May 2003.

[3] Jacobson, V., Michael J. Kar ”Congestion Avoidance and Control”, Proc. ACM SIGCOMM, Stanford, CA, ACM Computer Communications Review (CCR), Vol.18, No.4, 1988, pp.314-329.

[4] M. Allman, V. Paxson, W. Stevens, ”TCP Congestion Control”, RFC 2581, April 1999.

[5] Salahuddin Huhammad Salim Zabir and Gen Kitagata, ”A proposal for efficient TCP flow control over satellite networks”, Telecommunication systems 25:3, 4, 371-400, 2004.

[6] M.A. Alnuem, J.E. Mellor, R.J. Fretwell, ”Explicit Congestion Notification for error discrimination A practical approach to improve TCP performance over wireless networks ”, Wireless Telecommunications Symposium (WTS) , 2007.

[7] Rajesh Krishnan, James P.G. Sterbenz, Wesley M. Eddy, Craig Partridge, and Mark Allman, ”Explicit transport error notification (ETEN) for error-prone wireless and satellite networks,” Computer Networks (Elsevier), Volume 46, Number 3, Pages 343-362, 2004.

[8] Gustave Anderson, ”Bit-Error-Rate and Capacity Estimation in Wireless Networks”, PhD Thesis, Drexel University, March 2011.

[9] O. K. Tonguz, S. Maloo, A. Xhafa, and S. Sheeshia, ”Internet Access via LEO Satellite Networks: TCP/IP or ATM?”, IEEE Global Telecommunications Conference (GLOBECOM’99), pp. 301-305, Rio de Janeiro, Brasil, Dec. 1999.

[10] David Andrew Carek. ”Packet-Based Protocol Efficiency for Wireless Communications”, Journal of Aerospace Computing, Information, and Communication, Vol. 2, No. 5 (2005), pp. 238-251.

[11] D. Velenis, D. Kalogeras, B. Maglaris, ”SaTPEP: a TCP Performance Enhancing Proxy for Satellite Links”, in PROC 2nd International IFIP-TC6 Networking Conference, Pisa Italy, May2002.

[12] S. Floyd, T. Henderson and A. Gurtov, ”The NewReno Modification to TCP’s Fast Recovery Algorithm”, RFC 3782, April 2004.

[13] B. Constantine, G. Forget, R. Geib, R. Schrage ”Framework for TCP Throughput Testing”, RFC 6349, August 2011.

[14] T. Vu, D. Reschke, W. Horn, Dynamic packet size mechanism (DPSM) for multimedia in wireless networks, in: Proc. of Multimediale Informations- und Kommunikations systeme (MIK), Erfurt, Germany, September 2002.

[15] S. Floyd, T. Henderson, ”The NewReno Modification to TCP’s Fast Recovery Algorithm”, RFC 2582, April 1999.

[16] ”Network Simulator (NS-2)”, http://www.isi.edu/nsnam/ns/

[17] ”SaTPEP model for NS2 simulation”, http://www.netmode.ntua.gr/ dbelen/SaTPEP/.

[18] ITU-R, ”Multi-carrier based transmission techniques for satellite systems”, Recommendation ITU-R S.2173, 2011.

[19] Bruno Perrot, ”Satellite Transmission, Reception and On-Board Processing Signaling and Switching”, Handbook of Satellite Applications, 2013, pp. 359-372.

[20] Bernd Friedrichs, Philipp Wertz, ”Error-Control Coding and Packet Processing for Broadband Relay Satellite Networks with Optical and Microwave Links”, Advanced Satellite Multimedia Systems Conference (ASMS) and 12th Signal Processing for Space Communications Workshop (SPSC), 2012.

[21] Zhang, Qixian Zhang, Sihai Zhou, Wuyang, ”Enhanced LT decoding scheme in satellite communication”, Sixth International Conference on Wireless Communications & Signal Processing (WCSP), IEEE 2014, pp.1-6.

**J. Govindarajan** received the M.Tech in Software engineering from Anna University, Chennai, India, in 2006. He has 10 years of teaching experience after his master’s degree. He is currently working as Assistant Professor in Dept.of CSE, Amrita Vishwavidyapeetham University, Coimbatore, Tamil Nadu , India. His current research interests include optimization of transport protocol for satellite network, Adhoc Network.

**G. Kousalya** received Ph.D in Computer Science and Engineering from Anna University, Tamil Nadu, India. She has more than 20 years of experience in teaching. She is working as Professor in Department of CSE, Coimbatore Institute of Technology, Coimbatore, India. Her research interests include Wireless Networks and Distributed Systems.