INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN ENGINEERING AND TECHNOLOGY (IJARET)

ISSN 0976 - 6480 (Print)
ISSN 0976 - 6499 (Online)
Volume 5, Issue 4, April (2014), pp. 131-140
© IAEME: www.iaeme.com/ijaret.asp
Journal Impact Factor (2014): 7.8273 (Calculated by GISI)
www.jifactor.com

IJARET
©IAEME

ENHANCED TCP HYBLA AND PERFORMANCE COMPARISON FOR SATELLITE LINK

Rakesh D. Vanzara[1], Dr. Haresh S. Bhatt[2]

[1]Department of Information Technology, U V Patel College of Engineering, Mehsana, Gujarat, India
[2]Space Applications Centre, Indian Space Research Organization, Ahmedabad, Gujarat, India

ABSTRACT

Satellite link can be used to bridge the gap where traditional terrestrial network cannot reach or feasible. It always provides the connectivity even at the time of natural disaster. Satellite link has very high round trip time (RTT) and high bit error rate (BER). Standard transport protocol like TCP performs poorly for the satellite link. TCP hybla was proposed to resolve the issue of high RTT for the satellite link but its performance get degraded in presence of link losses. TCP westwood was proposed to resolve the issues of link losses for wireless networks but it performs poorly for the link with high RTT. In this paper, we present the detailed performance comparison of TCP hybla, TCP westwood along with proposed enhanced TCP hybla which improves the performance significantly for the satellite link. Enhanced TCP hybla combines the approach of TCP westwood with the TCP hybla along with few must needed modifications. This enhanced approach is implemented and tested using NS 2.

Keywords: Enhanced TCP Hybla, High BER, High RTT, Satellite Link, Wireless Networks.

I. INTRODUCTION

Standard TCP protocols like NewReno [1, 2] give optimized performance for wired networks. These protocols increase the congestion window (cwnd) by one on receipt of an acknowledgment in slow start (ss) phase and increase the cwnd by one per RTT in congestion avoidance phase. The rate at which acknowledgments are received depends on RTT of the connection. Hence, higher the RTT, lower the cwnd increase rate. If cwnd increase rate is low, the link capacity cannot be utilized effectively. Compared to wired link, the satellite link has very high RTT. Hence, the performance of TCP on satellite link is significantly low. Standard TCP assumes congestion by means of packet loss. Packet loss is indicated either by 3 duplicates acknowledgements or timeout. Packet loss results into reduction of cwnd. In a link like satellite a packet loss can also occur due to the link error. As a result, standard TCP cannot differentiate the loss and reduces the cwnd in case of link error. Hence, such unnecessary reduction of cwnd worsens the performance further. The remaining paper is organized as follows. Section 2 describes the related work for the high RTT and high BER. Section 3 presents the enhanced hybla approach in details. Section 4 describes the simulation topology and detailed comparison of TCP hybla, TCP westwood and enhanced TCP hybla by means of simulation results in detail. Finally, section 5 concludes the paper.

II. RELATED WORK

TCP westwood [3] was proposed to solve the issue of loss discrimination on wireless link. It does so by continuous estimation of the bandwidth used by the connection. If packet loss occurs, it sets ss threshold and cwnd according to the estimated bandwidth. It solves the issue of packet error rate and improves the performance on wireless link. In presence of high RTT, TCP westwood performs poorly as it was not designed for high RTT link. Hence, the performance of the TCP westwood is significantly low in presence of high BER and High RTT link. TCP hybla [4] was proposed to resolve the problem of high RTT for the satellite link. TCP hybla tries to achieve the same data sending rate as of reference wired connection. It does so by making cwnd increase rate independent of the RTT. TCP hybla uses the normalized RTT to set the ss threshold and cwnd after the congestion episode. In this way, this protocol is able to increase the cwnd very fast even for the high RTT link. Thus, TCP hypla is able to utilize the available bandwidth for the link like satellite with high RTT. Performance of TCP hybla gets reduced in the presence of high BER as it is not designed to perform well in presence of high BER.

In this paper, we propose the modifications to TCP hybla to resolve the problems in performance due to high RTT and high BER of satellite link. Enhanced TCP hybla is implemented in NS-2 [5] using NS2-TCP-Linux patch [6, 7, 8]. Using this patch, we can use the linux TCP congestion control modules within the NS-2.

III. ENHANCED TCP HYBLA APPROACH

A. Congestion Window Update

Enhanced TCP hybla cwnd update rule is same as TCP hybla in ss and congestion avoidance (CA) phase. Standard TCP in ss initializes the cwnd with one segment and increases it by one on receiving an acknowledgment for the same. In CA phase, it increases cwnd by one per RTT if all the segments are acknowledged. TCP hybla equalizes the data sending rate of satellite link (with RTT) with reference to wired connection (with low RTT link). It calculates the normalized round trip time ρ as:

$$\rho = RTT/RTT0 \qquad (1)$$

RTT0 is round trip time of reference wired connection to which TCP hybla tries to equalize the data sending rate. TCP hybla's cwnd update rules are [4]:

$$\begin{aligned} WH_{i+1} &= WH_i + 2^{\rho} - 1, \quad SS \\ &= WH_i + \rho^2/WH_i, \; CA \end{aligned} \qquad (2)$$

In above equations, WHi is the cwnd when i^{th} acknowledgment is received. When TCP hybla operates in ss phase and receives an acknowledgment, cwnd is increased by $2^{\rho} - 1$ and in CA phase by ρ^2/WH_i. Enhanced TCP hybla updates the values of cwnd in slow start and congestion avoidance phases as per the equations listed above.

B. Bandwidth Estimation & Congestion Episode

TCP hybla assumes the congestion in network due to the loss of the segments. Segment loss is detected by means of either 3 duplicate acknowledgments or when timer goes off. In case of 3 duplicate acknowledgments, it sets ss threshold to half of the current cwnd and cwnd is updated to the one fourth of the current cwnd. When timer goes off, it sets ss threshold value to the same as in case of 3 duplicate acknowledgments, but cwnd is set to ρ. Thus, TCP hybla assumes that the loss is due to the congestion only and so it performs poorly on link like satellite where loss can be also due to the link characteristics. Major contribution of this paper is that in enhanced TCP hybla, we are able to increase the cwnd significantly even after the loss of segments which is due to the link error. This results into significant performance improvements in presence of link losses, which is the usual scenario for the satellite link.

Enhanced TCP hybla uses the indirect way to differentiate the reason for the loss of the segment. In case, the loss is due to the link characteristics, cwnd needs to be increased rather than decreasing it. To achieve this objective, enhanced TCP hybla tries to estimate the utilized bandwidth. To estimate the utilized bandwidth, of the link, enhanced TCP hybla counts the received segments by measuring the number of acknowledgments received. After that, it calculates the utilized bandwidth by dividing received bytes by the time period for which acknowledgments were counted. After detecting a loss of segment (which can be triggered by either 3 duplicate acknowledgments or time out), new ss threshold and cwnd are set as per the following algorithm:

```
IF (3 DUPACK received)
        IF(BWE <  CC/2)
          Residual pipe capacity = (CC - BWE)*RTTmin/seg_size
          SSthreshold = MAX(Residual pipe capacity,  0.75*packets_in _flight)
          cwnd = MAX(Residual pipe capacity,  0.75*packets_in _flight)
        Else
         Minimum pipe capacity = (BWE * RTTmin) / seg_size
         SSthreshold = MAX(Minimum pipe capacity,  0.75*packets_in_flight)
        cwnd = MAX(Minimum pipe capacity,  0.75*packets_in_flight)
        End If
EndIf
```

In above algorithm, channel capacity (CC) is bottleneck link capacity. BWE is the estimated bandwidth used by the connection. RTTmin is the minimum RTT observed over the time, seg_size is the segment size at the transport layer and packets_in_flight is the number of segments currently in flight. As per the above algorithm, after receiving 3 duplicate acknowledgments, if the bandwidth used by connection is less than half of the actual capacity of the link, ss threshold and cwnd are set according to the difference between the actual capacity of the link and bandwidth used by the connection. This is the major modification compared to TCP westwood. TCP westwood uses the minimum pipe capacity to set the value of ss threshold and cwnd after detecting the loss by means of 3 duplicate acknowledgments. This results into significant performance improvement.

If bandwidth used by the connection is greater than half of the actual capacity of the link, bandwidth used by connection is used to set the value of ss threshold and cwnd. Hence, when used bandwidth is less, connection is aggressive. When timeout occurs, enhanced TCP hybla uses the same algorithm as described above except cwnd is set to ρ.

IV. PERFORMANCE EVALUATION AND COMPARISON

A. Simulation Topology

Performance evaluation is carried out using NS-2 [5] for the topology shown in Fig. 1. Enhanced TCP hybla code is written for the NS-2-TCP-Linux patch [6]. This patch provides the facility to use the congestion control modules of linux kernel in NS-2 which works as bridge between NS-2 and linux congestion control module.

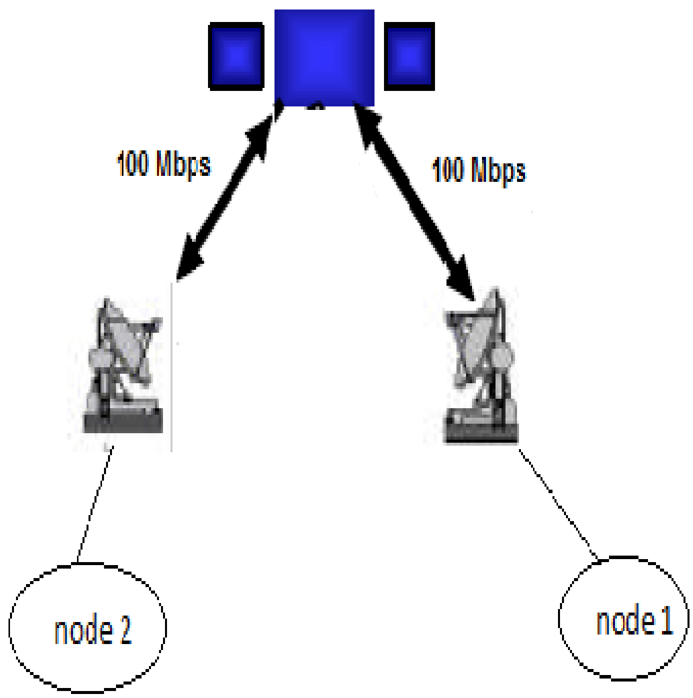

Figure 1: Simulation Topology

As shown in Fig.1, satellite link is configured to have 100 Mbps link [9] of uplink and downlink and RTT is set to 600 ms. In all simulations we have assumed the packet size is of 1500 bytes. In this paper, the aim is to test the performance of the protocols in presence of high RTT and losses due to the link characteristics. Therefore, the performance evaluation is carried out for different values of BER to analyze its effects on three protocols discussed in this paper.

B. Results & Discussion

In this section, performance of the enhanced TCP hybla is compared with TCP hybla & TCP westwood and discussed in detail. In Fig. 2, X axis shows time in seconds and Y axis shows the cumulative goodput in Mbps of the connection. Cumulative goodput is the number of correctly received bits over the time. Fig. 2 represents the situation where there is no loss at all. Thus, in case of TCP hybla, it starts as the standard TCP, but its ss threshold value is set as per the initial estimation of the bandwidth which results into significant performance improvement in the initial few seconds. As TCP hybla's ss & CA phases are driven by normalized round trip time, it results into fast increments of the cwnd. Hence, it starts operating at 96 Mbps within 100 seconds of the simulation time.

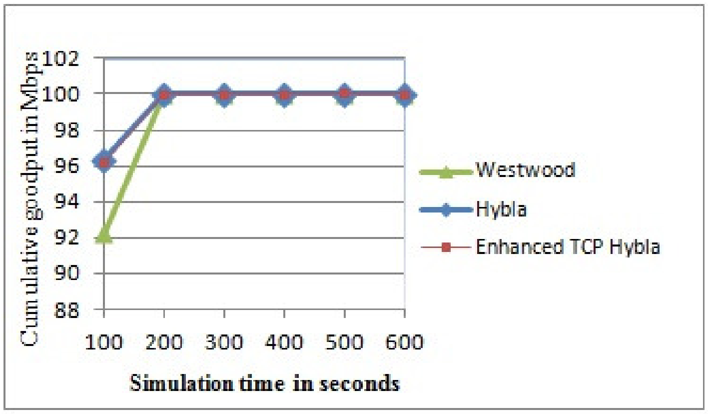

Figure 2: Performance for BER=0

Same result is quite apparent for the enhanced TCP hybla because it is the no loss scenario. TCP westwood start with ss and it takes longer time to increase the cwnd as its increase rate depends on the flow of the acknowledgments. As TCP westwood is not specifically designed for the high RTT link, it takes longer time to operate at the peak of the available bandwidth.

As shown in Fig. 3, performance of TCP westwood remains below 50 Mbps for the BER of 10^{-8} and RTT of 600 ms. The reason for such a performance is that westwood increases its cwnd based on the flow of acknowledgment and set the ss threshold and cwnd based on the utilized bandwidth after the event of an error. TCP hybla also operates around 50 Mbps but due to a loss, it reduces the cwnd as it is not using any bandwidth estimation technique to set the cwnd and ss threshold after the event of a loss. Hence, its cumulative goodput gets reduced after the error and again increases the cwnd which increases the cumulative goodput. As shown in Fig.3, in presence of errors, enhanced TCP hybla gives the same performance as it is the no loss scenario (i.e. 0 BER).

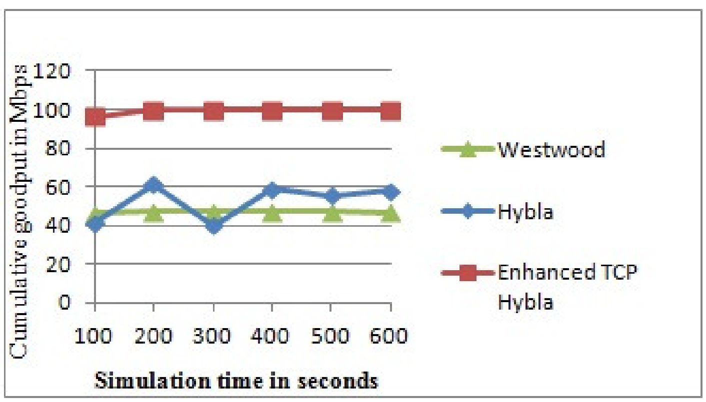

Figure 3: Performance for BER= 10^{-8}

As enhanced TCP hybla uses the residual pipe capacity rather than minimum link capacity (used in westwood) to set the value of cwnd and ss threshold in case the utilized bandwidth is less than half of the actual capacity. Hence, it is able to increase the sending rate very fast. Compared to TCP hybla, enhanced hybla indirectly decides that the loss is due to the link error. Hence, it does not reduce the cwnd and results into the same performance as scenario of 0 BER.

Fig. 4 shows the performance for the BER of 10^{-7}. TCP westwood follows the standard approach in ss and CA phase and it is using only utilized bandwidth to set the cwnd. Thus, in this scenario as BER is high compared to previous two scenarios, due to the losses, its cwnd remains low which in turns keep utilized bandwidth low. TCP westwood sets the SS threshold and cwnd based on the utilized bandwidth after the losses. As value of cwnd decides the amount of data the sender can send, which in turns decide the goodput. Hence, goodput of the TCP westwood remains around 17 Mbps for 100 Mbps link at 600 seconds, which indicates link utilization of 17% only. TCP hybla keeps reducing the cwnd and ss threshold on the event of a loss and hence, it can only utilize the 12 Mbps out of the 100 Mbps link.

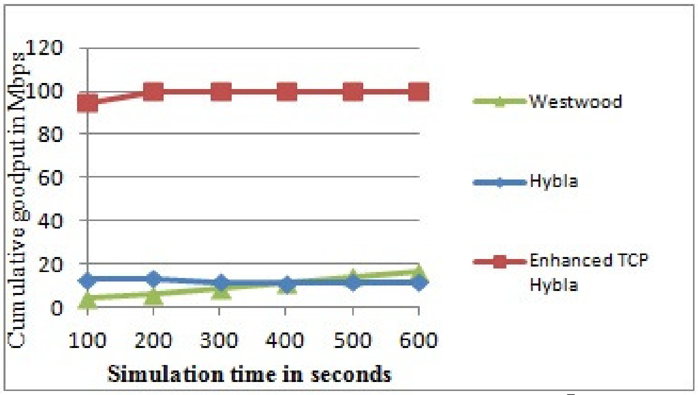

Figure 4: Performance for BER = 10^{-7}

It is quite apparent from Fig. 4 that enhanced TCP hybla is able to increase the cwnd and ss threshold even after the losses. It means that enhanced TCP hybla approach is able to deduce that loss is not the indication of the congestion but due to the link error. This proposed enhancement uses the residual capacity of the link to set the cwnd and ss threshold when data sending rate is low and uses the utilized bandwidth when the data sending rate is high. In this way, enhanced TCP hybla is able to achieve the goodput as high as the capacity of the link in presence of high BER and high RTT.

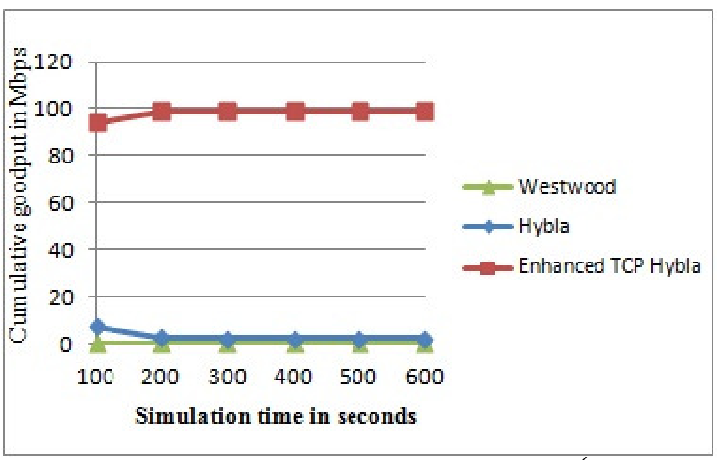

Figure 5: Performance for BER = 10^{-6}

Fig. 5 & 6 shows the performance for the BER of 10^{-6} and 10^{-5} respectively. As we can see from the graph, TCP westwood is giving a goodput less than 0.5 Mbps for the link of 100 Mbps. The reason for such a poor performance is that due long RTT it cannot increase the cwnd fast and due to high BER, after losses it sets the ss threshold and cwnd as per the estimated utilized bandwidth which is obviously very low.

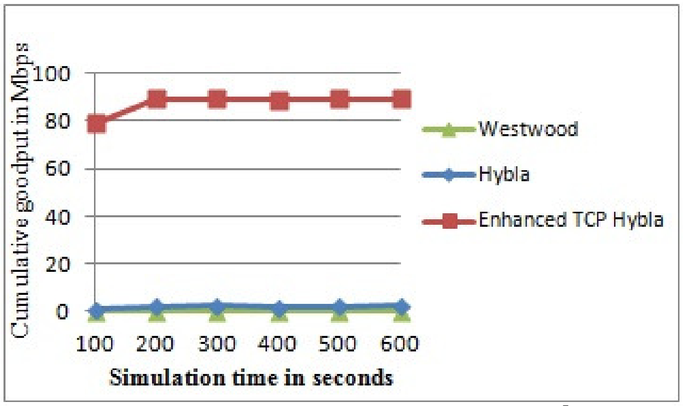

Figure 6: Performance for BER = 10^{-5}

For the performance of TCP hybla the same reasons are applicable as discussed for the scenario of BER of 10^{-7}. As shown in Fig. 5 & 6, TCP hybla performs poorly compared to enhanced TCP hybla. The reason for such a poor performance is that it does not have the sense to decide the reason for the losses. Hence, it keeps reducing the cwnd & ss threshold blindly in presence of link losses.

Enhanced TCP hybla is able to operate at the peak of the link capacity even for the BER of 10^{-6} as shown in Fig. 5. It is reaching the peak value at around 200 seconds. In Fig. 6, performance of westwood is less than 0.1 Mbps and hybla operates around 2 Mbps. Performance of enhanced TCP hybla will remains significantly high compared to the westwood and existing hybla approach. Other observation from Fig. 6 is that enhanced TCP hybla start operating around 90 Mbps after 200 seconds. The reason for such behavior is that due to the initial losses and timeout, cwnd remains low compared to BER of 10^{-6} or less. Thus, lower value of cwnd results into lower data sending rate and in turns the goodput is low.

C. Other Results & Discussion

In this section, we have presented the results for the three protocols for various values of RTT, BER and also compared the results for different file size data transmission. Simulation topology used for the results is the same as shown in Fig.1. As shown in Fig. 7, RTT is varied from 100 ms to 600 ms for the satellite link and cumulative goodput is measured at BER of 10^{-8}. Results show that performance of enhanced hybla is not affected by the change in RTT for BER of 10^{-8}, while hybla operates around 50 Mbps. Westwood performance decreases as RTT is increased.

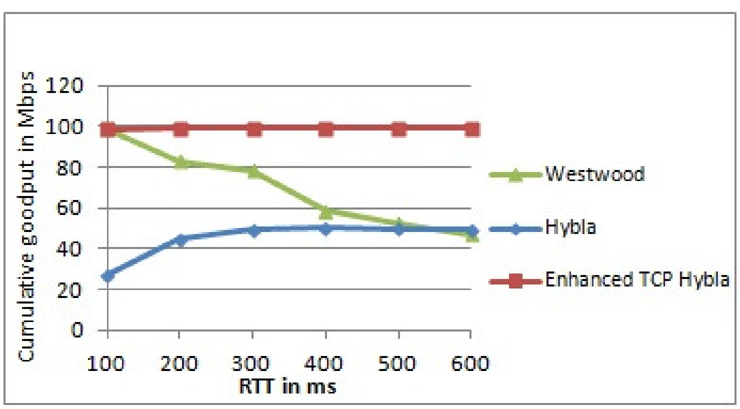

Figure 7: Effect of RTT on goodput for BER = 10^{-8}

As shown in Fig. 8, performance of TCP hybla and enhanced TCP hybla is same for the BER of 10^{-10} or less. As BER is increased, performance of hybla and westwood degrades significantly while enhanced hybla performance is as expected. Hence, from the Fig. 8, it is very clear that hybla can perform well with high RTT but performs poorly as BER is increasing and westwood performs better in presence of losses but performs poorly for the link with high RTT. Thus, enhanced TCP hybla can be a very useful protocol for the link like satellite where RTT is also very high and BER is non negligible.

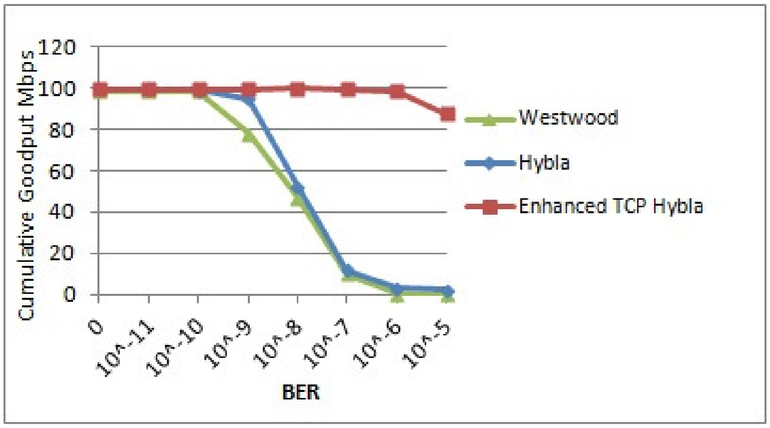

Figure 8: Effect of BER on goodput for satellite link with RTT=600ms

Fig. 9 shows the performance for FTP application with file size of 50 KB to 2 GB for the BER of 0. As file size is increased performance get increased as expected in the case of no loss scenario. Westwood operates at the peak of the bandwidth for 2 GB file while hybla and enhanced hybla start operating near to the peak of the bandwidth for the file size of 500 MB.

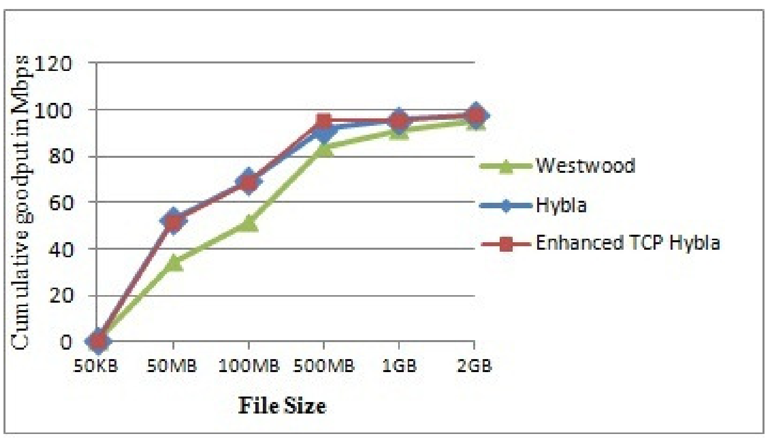

Figure 9: Performance for different file size for satellite link, RTT=600ms, BER=0

Fig. 10 shows the performance for FTP application with file size of 50 KB to 2 GB for the BER of 10^{-8}. Westwood is able to achieve the goodput of 50 Mbps which again proves that in presence of high RTT and high BER it cannot utilize the available bandwidth fully. Hybla operates at 53 Mbps for 100 MB file but for 500 MB file transmission, its cumulative goodput is 41 Mbps. The reason for such a poor performance is that, as data transmission size gets increased more errors would be there and in turns on each error hybla keeps reducing the cwnd. The same is true for the 1 GB and 2 GB file transmission but in this case, hybla takes more time to transmit and able to increase the cwnd significantly during the simulation time which results into cumulative goodput of around 50 Mbps. Enhanced TCP hybla is giving the same performance as in case of BER 0.

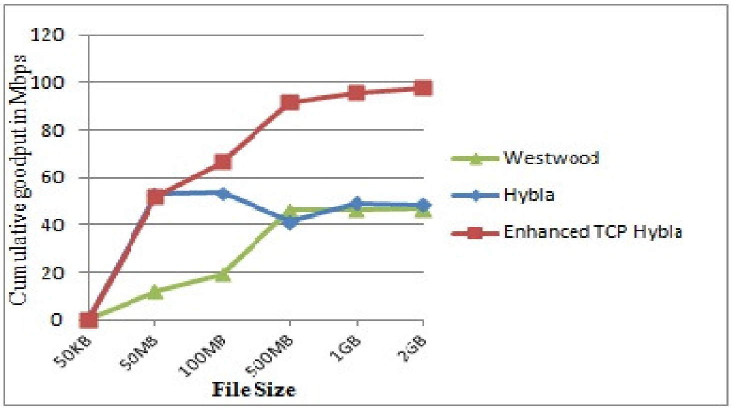

Figure 10: Performance for different file size for satellite link, RTT=600ms, BER=10^{-8}

Fig. 11 shows the performance for BER of 10^{-7}. Westwood can achieve maximum goodput of 10 Mbps for 2 GB file and so transmission time is very high. Hybla cannot perform well even for the file size of 100 MB and more and goodput is only 11 Mbps. Thus, for BER of 10^{-7} and file size of 1 GB and 2 GB transmission time is significantly high. The reason for such a poor performance is its inability to with stand with the high BER. For the file size of 500 MB or more enhanced TCP hybla is able to achieve the throughput of 90% and more. When we send the more amounts of data enhanced TCP hybla is able to explore the available bandwidth even in presence of significant bit error rate as high as 10^{-7}.

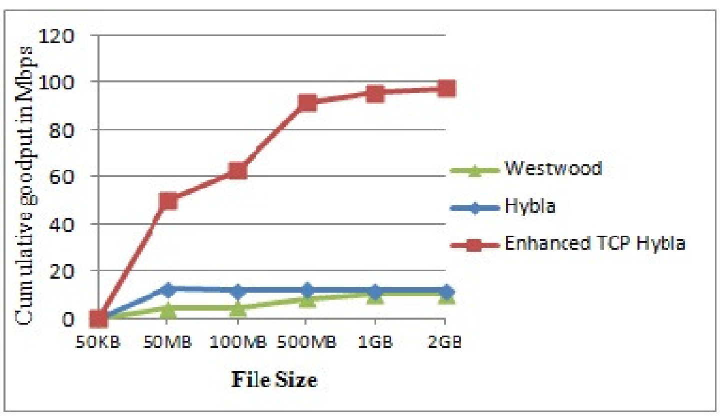

Figure 11: Performance for different file size for satellite link, RTT=600ms, BER=10^{-7}

V. CONCLUSION & FUTURE WORK

In this paper, the detailed comparison of westwood, hybla and proposed enhanced TCP hybla is presented. The objective of this enhancement is to have a protocol which can withstand in high RTT & high BER environment which is the common characteristic of the satellite link. Simulation results show that there is a significant improvement in the performance of enhanced TCP hybla compared to TCP hybla and TCP westwood. In future, the proposed protocol performance on the real environment can be tested. We are further planning to improve the initial behavior of the enhanced TCP hybla in presence of high BER.

REFERENCES

[1] R. Wang, T. Taleb, J.Abbas, Bo Sun "protocols for reliable data transport in space Internet", *IEEE Communications surveys & Tutorials*, vol. 11, pp. 21 – 31 Second Quarter 2009.

[2] Floyd S, Henderson T. The NewReno modification to TCP's Fast recovery algorithm. Request for Comment 2582, April 1999, IETF.

[3] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, "TCP westwood: Bandwidth Estimation for enhanced Transport Over Wireless Links"," ACM Mobicom 2001",vol 8 issue.5, pp- 287–297, july 2001.

[4] C. Caini and R. Firrincieli, "TCP Hybla: A TCP enhancement for heterogeneous networks," *Int. J. Satellite Commun. Netw.*, vol. 22, pp.547–566, Sep.–Oct. 2004.

[5] NS-2, Network Simulator, www.isi.edu/nsnam/ns

[6] D. X. Wei and P. Cao, "NS-2 TCP-Linux: An NS-2 TCP implementation with congestion control algorithms from Linux," presented at the Value Tool Workshop NS-2, Pisa, Italy, 2006.

[7] D. X. Wei and P. Cao, "A Linux TCP implementation for NS2", Manual for patch, May 2006.

[8] D.X. Wei and P. Cao, "A mini-tutorial for TCP –Linux in NS- 2", May 2006.

[9] Haresh S. Bhatt, Hitesh J Kotecha, VH Patel, K. Bandyopadhyay, "GridTCP: A transport layer data transfer protocol for satellite based Grid Computing", Proceedings of WoNGeN'05, International workshop on Next Generation Wireless Networks, Goa,India, 18-21 Dec, 2005.

[10] Hu Y, Li VOK., ''Satellite-based Internet: a tutorial.'' *IEEE Communications Magazine*, pp.154-162, 2001.

[11] I.F.Akyildiz , G.Morabito, S.Palazzo, "Research issues for transport protocols in satellite IP networks.", *IEEE Personal Communications* , pp. 44–48, 2001.

[12] C.Barakat, E.Altman, W.Dabbous, "On TCP performance in a heterogeneous network: a survey.", *IEEE Communications Magazine,* pp. 40–46, 2000.

[13] T.Henderson, R.Katz. "Transport protocols for Internet-compatible satellite networks.", *IEEE Journal on Selected Areas in Communications (JSAC)*, pp. 326–344, 1999.

[14] C.Roseti, F.Zampognaro, M.Luglio, "Enhancing TCP Performance over Hybrid Wireless Terrestrial-Satellite Networks", *First International Conference on Advances in Satellite and Space Communications-SPACOMM 2009*, pp.19-23, July 2009.

[15] Boney Bose Kunnel, Susan Abraham, R Kumar and Swarna Ravindra Babu, "A Comparative Study on Spectral Analysis of Global Navigation Satellite Systems", *International Journal of Electronics and Communication Engineering & Technology (IJECET)*, Volume 4, Issue 2, 2013, pp. 390 - 398, ISSN Print: 0976- 6464, ISSN Online: 0976 –6472.

[16] Dheyaa Jasim Kadhim and Sanaa Shaker Abed, "Performance and Handoff Evaluation of Heterogeneous Wireless Networks (Hwns) Using Opnet Simulator", *International Journal of Electronics and Communication Engineering & Technology (IJECET)*, Volume 4, Issue 2, 2013, pp. 477 - 496, ISSN Print: 0976- 6464, ISSN Online: 0976 –6472.

[17] Mazin Ali A. Ali, "Characterization of Fog Attenuation for Free Space Optical Communication Link", *International Journal of Electronics and Communication Engineering & Technology (IJECET)*, Volume 4, Issue 3, 2013, pp. 244 - 255, ISSN Print: 0976- 6464, ISSN Online: 0976 –6472.