Round-Trip Time and Available Bandwidth Estimation Based Congestion Window Reduction Algorithm of MPTCP in Lossy Satellite Networks

Mohmmad Abu Yousuf1,*, Md Mazidul Islam2, Md Shawkat Hosen3 and Md Liton Ali4
1,2,3,4School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China

*Corresponding author email: L201720055@stu.cqupt.edu.cn

Abstract. A significant enhancement in recent communication systems is Satellite Communications. The performance improvement of the transmission control protocol (TCP) in satellite communications has been an active research area. Multipath TCP (MPTCP) ensures a robust solution than the regular TCP. However, the performance degradation in satellite networks due to the link asymmetry, high loss and high latency cannot distinguish the packet losses caused by network congestions from the losses properly. In this paper, a round-trip time detection-based algorithm is proposed using available bandwidth estimation (ABE) to mitigate the issue of blind decreasing of congestion window upon detecting the packet loss over lossy satellite networks. Experiments are done using the NS-2 network simulator to compare the various congestion control algorithms of both TCP and MPTCP over satellite networks. Simulations results from the experiments show that our proposed algorithm outperforms the existing algorithms which are mentioned in this article.

Keywords: Lossy satellite networks; MPTCP; Congestion control; ABE; Round-Trip time.

1. Introduction
Satellite communication networks have been progressed significantly in recent decades [1]. Mobile terminals are equipped with multiple interfaces to mitigate packet loss and to ensure reliable communication. As a result, TCP [2] extends to MPTCP, which can utilize various interfaces at the same time [3]. Additionally, MPTCP is compatible with regular TCP, which makes it convenient to deploy in real networks. If any end host does not support MPTCP, it acts as a regular TCP as well. There are four fundamental transmission phases of TCP transmission, namely slow start (SS), congestion avoidance (CA), fast retransmit, and fast recovery [4]. It maintains two main variables, i.e., the congestion window (cwnd), which is initially set to be 1 maximum segment size (MSS), and the slow-start threshold (ssthresh). At the beginning of the TCP connection, the sender enters in the SS phase, in which it increases the cwnd by 1 MSS for each ACK (Acknowledgements) it receives. Therefore, in the SS phase, the TCP sender’s cwnd grows exponentially. When the cwnd exceeds the ssthresh value, the CA phase is initialized. During this phase, cwnd increases by 1 MSS for every ACK. It is equivalent to an increment of by 1 MSS for every round-trip time (RTT). Upon detecting packet loss in the network, the sender receives a duplicate ACK (DUPACK) from the receiver side. When three consecutive DUPACKs is received, the network is considered as a congested one, and TCP decreases the cwnd by half of the current cwnd size. This additive increase and multiplicative
decrease method of cwnd is known as AIMD [5]. It leads to the linear growth of the transmission rate that helps the sender to utilize the available network bandwidth. Most of the congestion control algorithms of MPTCP also follows the AIMD method like TCP such as Coupled [6], LIA (Linked Increased Algorithm), OLIA (Optimized Linked Increased Algorithm) [7] etc. These algorithms interpret packet loss as congestion like a regular TCP. As a result, the congestion control algorithm reduces the congestion window to half and sets the ssthresh, whether it is congested or not, which leads to very low bandwidth utilization. Though these algorithms work well for wired networks, but the performance decreases a lot as for the long round-trip time and high packet loss rate of wireless or lossy satellite links [8]. As a consequence, some approaches have been taken to mitigate these issues in TCP Westwood [9], TCP Jersey [10], TCP wVegas [11], for single flow in wireless networks. These algorithms have some drawbacks to satellite networks (Vegas is low jitter sensitive; Westwood and Jersey cannot distinguish random losses; Jersey does not follow standard behaviour of explicit congestion notification and unable to perform well for reverse packet transmission). As well as bandwidth and RTT detection-based algorithm [13] and [14] have also been proposed in MPTCP for satellite networks.

In this paper, we have proposed a congestion window reduction algorithm based on average RTT and available bandwidth estimation using LIA of MPTCP. Moreover, we have conducted two kinds of experiments. One is for single flow regular TCP using satellite networks while subflows are using the regular TCP, e.g., TCP Reno, Westwood, and Linux Highspeed in terms of the congestion window and sequence number increments. The other one is for MPTCP with two subflows for two different paths (satellite/terrestrial links) while using TCP in shared bottleneck link and finally, compared LIA with our proposed algorithm.

The rest of this article is illustrated as follows: In section 2, the overview of congestion control algorithms which have inspired us and our proposed round-trip time detection-based scheme is derived. Section 3 is devoted to evaluating the performance where simulation setup and results are presented for single and multipath flows. Finally, section 4 concludes this article with future work.

2. Overview and Congestion Control in Lossy Satellite Networks

In this chapter, the related well-performed TCP variants with their lacking are described, and as a solution to this, our algorithm is proposed.

2.1. Available Bandwidth Estimation and Optimized Congestion Window

TCP Jersey [10] is a variant of TCP congestion control for mixed wired and wireless networks. TCP Jersey can distinguish the causes of packet losses. It implements the same idea of TCP-Westwood to estimate the available bandwidth by monitoring the rate of the returning ACKs. Still, TCP-Jersey’s ABE provides a more accurate estimation which is executed from time sliding window estimator. When the nth ACK arrives at time $t_n$, the ABE is:

$$B_n = \frac{RTT \times B_{n-1} + L_n}{(t_n - t_{n-1} + RTT)}$$  \hspace{1cm} (1)$$

Where, $B_{n-1}$ is the previously estimated bandwidth at the time $t_{n-1}$, $L_n$ is the size of data that $n$th ACK acknowledges and the end-to-end delay is RTT at the time $t_n$. The optimum congestion window (ownd) in units of the segment is then calculated as

$$ownd = \frac{RTT \times B_n}{seg\_size}$$  \hspace{1cm} (2)$$

TCP Jersey supports ECN (Explicit Congestion Notification) implementation at the router, which is the extension of RED (Random Early Detection). ECN is an explicit signalling mechanism intended to convey network congestion information from intermediate routers to end nodes. It uses CW (congestion warning) scheme to distinguish the packets loss. The reason of using CW is that if the sender receives DUPACK due to packet loss with CW mark, TCP sender assumes the link is in a
congested state otherwise the loss has occurred because of transmission errors. Depending on them, it deploys the cwnd decreasing method. But it is unable to perform well to distinguish the random packet losses and to reverse packet transmission, as well as a router, has to perform extra works.

2.2. Congestion Control of Multi-path TCP

For MPTCP connections, sending rate of each subflow is required to maintain. There are several approaches to MPTCP congestion control [12]. LIA is one of the default implementations of congestion control algorithms of MPTCP. It is a coupled congestion control algorithm which dynamically controls the aggressiveness. Considering $R$ the set of subflows, each subflow $r$ may take a different route through the internet. Each subflow $r \in R$ maintains its congestion window $w_r$. Let $w_{total}$ and $\alpha$ be the total window size of all subflows and the aggressive factor. After receiving an ACK on subflow $r$, the congestion window increases as,

$$w'_r = w_r + \min\left(\frac{\alpha}{w_{total}}, \frac{1}{w_r}\right)$$  \hspace{1cm} (3)

For each loss on subflow $r$, MPTCP decreases cwnd by $w_r / 2$.

LIA follows the slow start and the congestion avoidance as like TCP to increase the congestion window. Also, it reduces the cwnd blindly the same as regular TCP. This is an issue that degrades the performance of MPTCP upon packet loss. Since satellite communication maintains long RTT, link asymmetry, high packet loss rate etc. So, it needs to be considered a convenient algorithm to reduce the window. As a consequence of this, we have proposed a window reduction algorithm based on RTT and available bandwidth below.

2.3. Round Trip Time and Available Bandwidth Detection-based Congestion Control Algorithm for Satellite Networks

MPTCP’s LIA is an outstanding congestion control algorithm where the fairness of each sub-flow is guaranteed. We have improved the window reduction algorithm to reduce the cwnd based on LIA by using average RTT and set the ssthresh by available bandwidth detection methods. The congestion control algorithm based on available bandwidth detection of TCP Jersey inspired us. We probe the RTT in every few seconds of simulation for subflow $r$ and record the average RTT as

$$RTT_{r_{\text{avg}}} = \gamma \times RTT_r + (1-\gamma)\times RTT_{r_{\text{avg}}}$$  \hspace{1cm} (4)

The window reduction algorithm is introduced below,

| Algorithm: Congestion Window Reduction Algorithm after receiving 3 DUPACKs. |
|---|
| **Input:** $RTT_{r_{\text{avg}}}$ defined by equation (4), $ABE$ defined in equation (1) |
| **Output:** SS phase or CA phase |
| 1: For each subflow then |
| 2: $RTT_r =$ currentRTT & $ABE =$ currentABE |
| 3: if dupacks == 3 && $RTT_r > RTT_{r_{\text{avg}}}(1+\delta/100)$ then |
| 4: set ssthresh to own, by equation (2) |
| 5: set cwnd to ssthresh value |
| 6: and select SS phase or CA phase with |
| 7: Fast Retransmit the loss packet |
| 8: else |
| 9: remain in the same phase |
| 10: end |
| 11: end |
| 12: return SS phase or CA phase |
The value of $\delta$ adjusts depending on the difference of packet loss rate. In the satellite network, packet loss does not necessarily indicate the network has congested. However, if the congestion episode (reception of 3 DUPACKs) takes place, the round-trip time $RTT_r$ increases significantly as a result. The congestion control algorithm reduces the ssthresh to half and sets the cwnd as well. As a consequence of this procedure, congestion window size reduces significantly and takes a long time to be adjusted again as the propagation delay is very long in satellite networks. So, our method does not reduce it blindly after receiving three DUPACKs, but it first determines if the $RTT_r$ increased to $1 + \delta\%$ of $RTT_{r,avg}$. If this condition is satisfied, the congestion window will be reduced by determining the ABE. The ssthresh will be set, and then cwnd will be set as current ssthresh value. Additionally, we are not relying on ECN of the router to get the congestion warning (CW) which is one of the fundamentals of TCP Jersey as well as TCP Westwood. This is one of the advantages of our algorithm. The details of the modified congestion control algorithm are described below with the flow chart in figure 1.

![Flow chart of the proposed congestion control algorithm.](image)

The beginning of packet transmission starts with the slow start phase. After exceeding the ssthresh value, if there is no loss or any congestion episodes, the congestion avoidance phase takes place. If there is any packet loss (esp. three DUPACKs), the congestion window reduction phase starts. It works by estimating the available bandwidth of the network to set the ssthresh after comparing the current RTT value with average RTT value. If this satisfies the algorithm, this means the network has congested and then ssthresh and cwnd will be set according to the current situation of bandwidth. As well as fast recovery and fast retransmit as like standard TCP recovers the loss. At the final step, if the cwnd is less or equal to ssthresh, the slow start phase will start. Otherwise, the congestion avoidance phase will take place.

3. Performance Evolution

3.1. Simulation Setup

We have implemented Multipath TCP in ns-2.35 network simulator to evaluate the performance of our proposed congestion control algorithm. The patch for MPTCP is available in [15]. A Linux Ubuntu 18.04 machine is used to operate the experiments. The network model and simulation scenarios which we have mentioned in section 3.2 have several elements: clients (sender), terrestrial node, GEO bent
pipe satellite, receiver and satellite links. As well as, we have used the default round-robin scheduler [16] for MPTCP.

We have considered a network where sender side nodes use TCP with FTP traffic. A PEP (Performance Enhancing Proxy) [17] router is placed to split the traffic between the satellite path and terrestrial node. MPTCP is applied in this router. First, we have simulated without MPTCP (a simple one-way using GEO link) to obtain the performance in terms of cwnd and sequence number. Secondly, in MPTCP case, subflow 1 transmits the packets through a GEO repeater (bent pipe) satellite to the other side terminal to PEP receiver, and subflow 2 transmits the packets through the terrestrial path. In both channels, the loss rate is considered 0.01. The link parameters are in table 1.

| Table 1. Link parameters. |
|---------------------------|
| **GEO Link**               | **Terrestrial Link** |
| Data Rate                 | 10 Mbps              | 2 Mbps              |
| Delay                      | 250 ms               | 50 ms               |
| Loss rate                  | 1 (%)                | 1 (%)               |

The parameters for simulation are listed below in table 2. And other links (e.g., sender to the router, router to gateway etc.) parameters (data rate, delay) are taken the same as 2 Mbps and 5ms.

| Table 2. Simulation parameters. |
|----------------------------------|
| **Parameters**                   | **Value**         | **Parameters**         | **Value**         |
| window_                          | 100000           | δ (%)                  | 30               |
| packetSize_                      | 960              | TCP Variants           | Reno, Linux and Westwood |
| Application                      | FTP              | MPTCP Variants         | Coupled (LIA)     |
| Ecn                              | 0                | Simulation Time (sec.) | 100              |

3.2. Simulation Scenario and Result

From figure 2, the senders are connected to receivers using two paths (satellite and Terrestrial). The GEO satellite acts as a repeater and transfers the packets to the receiver side router. MPTCP is placed into this router.

![Figure 2. MPTCP transmission using satellite and terrestrial links](image)

For the very first case, we have evaluated the cwnd size of a single TCP flow where a GEO satellite is connected with a sender and a receiver node. The same topology and same interface have been used for all of the TCP variants. From figure 3, we can distinguish the cwnd increment and decrement for different single-path TCP where TCP Westwood shows a more stable connection.
Figure 3. Congestion window vs time graph for single-path TCP in lossy satellite networks.

In figure 4, we have illustrated the sequence number increment vs time of these variants. These graphs show that Westwood performs well in terms of sequence number also.

Figure 4. Packet sequence number vs time graph of different TCP agents in lossy satellite networks.

In figure 5 and figure 6, it is shown that the performance increases in terms of cwnd and the sequence number of two TCP flow while sharing the bottleneck link with MPTCP in satellite networks.

Figure 5. Congestion window vs time using MPTCP with the shared bottleneck.

Figure 6. Sequence number increment using MPTCP with the shared bottleneck.
So, sharing the links with MPTCP does not degrade the performance of single TCP in lossy satellite networks. And coupled congestion control algorithm (LIA) shows better performance than uncoupled one when Westwood is taken as the agent. Finally, it is worth to compare our modified MPTCP with LIA as it shows better performance than the others. From figure 7, it is observed that the sequence number increases rapidly while using the modified version of MPTCP congestion control. Therefore, more bytes are transferring through the links than the LIA and individual TCP.

![Figure 7. Comparison of modified MPTCP and LIA congestion control.](image)

4. Conclusion and Future Work

In this paper, we evaluate the different congestion control algorithms in lossy satellite networks. We have proposed the congestion window reduction algorithm based on round-trip time detection using available bandwidth estimation to set the ssthresh upon the packet loss detection while the sender side receives three duplicate acknowledgements. Finally, we have used NS-2 simulator to simulate the algorithm, and the results are shown in terms of the congestion window and the sequence number of packets. We may conclude that some TCP variants work well in satellite networks, but MPTCP (LIA) provides significantly better performance. As well as it is observed that our method outperforms the multipath TCP in satellite networks. Future work would be to improve the communication network by using round-trip time detection-based algorithm considering satellite diversity case with multipath TCP.

References

[1] Misra D, Misra D K, Tripathi S P, “Satellite Communication Advancement, Issues, Challenges and Applications”. International Journal of Advanced Research in Computer and Communication Engineering, vol. 2, no.4, pp 1681-1686, Apr. 2013.
[2] Allman M, Paxson V and Blanton E, “TCP Congestion Control”, IETF RFC 5681, pp 1-18, Sep. 2009.
[3] Ford A, Raiciu C, Handley M, Bonaventure O and Paasch C, “TCP Extensions for Multipath Operation with Multiple Addresses”, IETF RFC 6824, pp 1-82, Jun. 2019.
[4] Stevens W, “TCP slow start, congestion avoidance, fast retransmit and fast recovery algorithms,” IETF RFC 2001, Jan. 1997.
[5] Chiu D and Jain R, “Analysis of the increase/decrease algorithms for congestion avoidance in computer networks,” J. Comput. Networks, vol. 17, no. 1, pp 1–14, Jun. 1989.
[6] Raiciu C, Handly M and Wischik D, “Coupled Congestion Control for Multipath Transport Protocols”, IETF RFC 6356, pp 1-12, Oct. 2011.
[7] Wischik D, Raiciu C, Greenhalgh A and Handley M, “Design, implementation and evaluation of congestion control for multipath TCP”, Proceedings of the 8th USENIX conference on Networked systems design and implementation, pp 1-14, 2016.
[8] Tropea M and Fazio P, “Evaluation of TCP Versions over GEO Satellite Links”, IEEE International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), pp 1-5, Sep. 2013.
[9] Mascolo S, Gerla M, Casetti C and Sanadidi M Y, “TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links”, Proceedings of the 7th annual international conference on mobile computing and networks, pp 467-479, Jan. 2001.

[10] Xu K, Tian Y, and Ansari N, “TCP Jersey for Wireless IP Communications” IEEE journal on selected areas in communications, vol. 22, no. 4, pp 747-756, May. 2004.

[11] Cao Y, Xu M, Fu X, “Delay-based congestion control for multipath TCP”, IEEE 20th International Conference on Network Protocols (ICNP), pp 1-10, Feb. 2013.

[12] Xu C, Zhao J, Muntean G M, “Congestion control design for multipath transport protocols: A survey”, IEEE communications surveys & tutorials, vol. 18, no. 4, pp 2948-2969, Apr. 2016.

[13] Hu H, Hengli and Liu Y, “Bandwidth and Round-Trip Time Detection based Congestion Control for Multi-Path TCP over Highly Lossy Satellite Networks”, IEEE Global Conference on Signal and Information Processing (GlobalSIP), pp 1072-1075, Feb. 2019.

[14] Li H, Wang Y, Sun R, Guo S and Wang H, “Delay-Based Congestion Control for Multipath TCP in Heterogeneous Wireless Networks”, IEEE Wireless Communications and Networking Conference Workshop (WCNCW), pp 1-6, Nov. 2019.

[15] The MPTCP patch for Ns-2, Available: https://code.google.com/archive/p/multipath-tcp/downloads.

[16] Barré S, “Implementation and assessment of modern host based multipath solutions”, Université catholique de Louvain, Louvain-la-Neuve, Belgium, 2011.

[17] Border J, Kojo M, Griner J, Montenegro G and Shelby Z, “Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations”, IETF RFC 3135, pp 1-45, Jun. 2001.