International Journal of Engineering & Technology, 7 (4.31) (2018) 363-367

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

An Enhanced Congestion Control Algorithm for LEO Satellite Networks

Fahrul Hakim Ayob[1]*, Shamala Subramaniam[2], Mohamed Othman[3], Zuriati Zulkarnain[4]

[1,2,3,4]Jabatan Teknologi Komunikasi dan Rangkaian
Fakulti Sains Komputer dan Teknologi Maklumat
Universiti Putra Malaysia (UPM)
Serdang, Selangor, Malaysia
**Corresponding author E-mail: fahrulhakim@gmail.com*

Abstract

Since TCP WestwoodNew is designed to be implemented in wired and wireless network environment, there are few drawbacks found when TCP WestwoodNew is implemented in the satellite network environment. For examples, the sender cannot fully utilize the available bandwidth because the rate of the congestion window increment in Slow Start phase of TCP WestwoodNew is rather slow. The other problem is, since packets losses often occur due to link errors in satellite environment, TCP WestwoodNew tends to decrease its throughput drastically without committing proper available bandwidth estimation. In this research, there is a potential for TCP WestwoodNew to be improved by increasing its throughput and implemented in satellite networks. In this research, we suggest that the congestion avoidance algorithm of TCP WestwoodNew to be modified. This modification aims to improve the performance of TCP flows by increasing its throughput while attempting to maintain packet delay and the percentage rate of packet drops from getting worse.

Keywords: *bandwidth throughput; congestion control algorithm; LEO satellite network; TCP WestwoodNew; TCP Westwood*

1. Introduction

Transmission Control Protocol (TCP) congestion control was devised to ensure the stability of networks and to maintain bandwidth allocation to be always fair and efficient. The TCP Congestion control clarifies the technique to prevent sender nodes from exceeding network capacity by adjusting its transmission rate [1]. Congestion control is developed as a tool in networks that prohibits congestion from happening or even to get rid of the congestion if it occurred [2]. Basically, there are two prime objectives of the congestion control. First, to maintain the network to operate pretty close to its rated capacity even when encounters extreme traffic congestion. Second, to provide a fair service in supporting various Internet application with various QoS requirements. In the network, normally congestion control can be implemented in two circumstances [3]. The first one is network-assisted congestion control. This approach is used by routers to monitor its queue size based on the network condition. The second one is virtual circuit connection-oriented end-to-end congestion control with reliable protocol which is implemented in TCP which is the focus of this research.

The concept of end-to-end reliable TCP connection which is offered at the transport layer is resemblance a virtual circuit telephone connection but with reliable digital data transmission rather than merely voice communication. The sending and receiving process in TCP allows the delivery of data in the form of stream of bytes. In TCP, the sender divides the stream of data into segments. Every segment is numbered in order to guarantee it works in a sequence manner and to ensure reliability. When a receiver receives the segments in sequence, it sends acumulative acknowledgement (ACK) notification packet in return to the sender, informing that all of the data has been properly received in sequence manner [4]. But if the segment is received out-of sequence, a negative ACK will be produced by the receiver to notify the sender about the mishap and expecting the correct sequence number of segment to be resent. If an outstanding segment is not acknowledged within a certain period of time which results in timeout, the sender will retransmit the unacknowledged segments.

Most traditional Internet application such as World Wide Web (WWW), FTP and E-Mail require the satellite Internet to provide a proper TCP as the transport layer protocol. However, since conventional TCP was developed for wired networks, it is difficult to obtain good performance in the wireless situation such as satellite network. The combination of long Round-Trip Times (RTT) and high packet loss rates over satellite networks contribute the degradation of TCP performance [5][6]. So, a TCP congestion control which targets both wired and wireless network environment especially for satellite communication in this case was extremely required [5][7][8]. During the last decade there are many congestion control algorithms which have been suggested to improve the classic Tahoe/Reno TCP congestion control. One of them is TCP WestwoodNew which has been introduced by Hagag in [3].

The paper is organized as follows. Section 2.0 discusses about related work which mainly focuses on TCP WestwoodNew which has been the base of this work. Section 3.0 explains the proposed algorithm. All the criteria to be evaluated in this experiment and its setting is discussed in Section 4.0. Section 5.0 is results and discussion. The conclusion is in Section 6.0.

2. Related Study

TCP WestwoodNew could be seen as an improvement to the previous TCP Westwood's congestion avoidance and retransmission timeout (RTO) algorithms. But, with the increasing popularity of Internet applications which implement UDP protocol such as VoIP and streaming high quality video with massive size of bytes transferring, the networks becomes more congested due to continuous and constantly traffic flow. The situation become worse when the TCP traffic flow that brings traditional applications such as email and web browsing are severely affected by this phenomenon.

Since TCP WestwoodNew is suggested to be implemented in wired and wireless network environment only. There are few drawbacks when TCP WestwoodNew is implemented in the satellite network environment. For examples, the sender cannot fully utilize the available bandwidth because the rate of the congestion window increment in Slow Start phase of TCP WestwoodNew is rather slow. The other problem is, since packets losses often occur due to link errors in satellite environment, TCP WestwoodNew tends to decrease its throughput drastically without committing a proper available bandwidth estimation. In this research, there is a potential for TCP WestwoodNew to be improved a little bit by increasing its throughput and implemented in wireless satellite networks. The details on the congestion control algorithms of TCP WestwoodNew are discussed in [3].

3. Enhancement of TCP Westwood New

The proposed enhanced TCP Westwood New can be divided into two parts. The first part is the proposed Slow Start and Congestion Avoidance Phases algorithm and the second part is the proposed packet losses phase algorithm.

3.1 Proposed Slow Start and Congestion Avoidance Phase

The proposed Slow Start and Congestion Avoidance phase increases its congestion window size *cwnd* by summing the window size of default TCP (Reno) *cwnd* and estimated window size $cwnd_{abe}$, which is obtained from estimated available bandwidth *BWE*. *ssthresh* represents the slow start threshold. Undoubtedly, this method can increase the TCP traffic flow rate in congested satellite network which overwhelmed by UDP traffic flow. Estimated window size $cwnd_{abe}$ is obtained from Equation (1) [9].

$$cwnd_{abe} = BWE \times RTT_{min} / segment\_size \quad (1)$$

In Equation (1), *RTTmin* and *segment_size* represent the minimum Round Trip Time and the Maximum Segment Size respectively. Furthermore, the estimated bandwidth *BWE* is derived from two pieces of information: the receiving interval of *ACK* and the size of received data *d* indicated by *ACK* in the Equation (2) [3][9].

$$BWE = d / ACKinterval \quad (2)$$

The algorithm of proposed slow start and congestion avoidance phases are shown as in Fig. 1. In Fig. 1, slow start phase algorithm is written between line 5 until line 8 whilst, congestion avoidance phase algorithm is written between line 11 until line 21. Like TCP WestwoodNew, the size of *cwnd* in the modified TCP WestwoodNew would be adjusted according to the network condition based on the *BWratio* which is the ratio between the current bandwidth (BWCurrent) and previous bandwidth (BWprevious).

```
1   Estimate BWE
2   Set BWE = Bwcurrent
3   BWratio = BWcurrent / BWprevious
4   //slow start phase
5   If (cwnd < ssthresh)
6       {       cwnd = cwnd + 1 ;
7               cwnd = cwnd + cwnd_abe ;
8       }
9   else
10  //congestion avoidance phase
11   if (cwnd ≥ ssthresh)
12       if (1.5 > BWratio ≥ 1) {
13           cwnd = cwnd + 1/cwnd;
14           cwnd = cwnd + (cwnd_abe /2);
15       }
16       else if (BWratio ≥ 1.5) {
17               cwnd = cwnd + 2/cwnd;
18               cwnd = cwnd+ cwnd_abe;
19       }
20           else if (BWratio < 1)
21               cwnd = cwnd + 0;
```

Fig 1: Proposed Algorithm for Slow Start and Congestion Avoidance phases

The proposed algorithm in Fig. 1 does not only modify the congestion avoidance phase of TCP WestwoodNew as in [3] but also modify the slow start phase too. This proposed algorithm increases the congestion window size quickly using both the window control of TCP WestwoodNew and the values based on the results of estimated bandwidth *BWE*. In the congestion avoidance phase, if *BWratio* is more than 1.5, this indicates that there is a lot of decrease in the network activity. Therefore, the congested window size *cwnd* should be increased with the estimated congested window size $cwnd_{abe}$ which is based on estimated available bandwidth *BWE*. If *BWratio* is more than 1 but less than 1.5, congested window should be increased by the half of estimated congested window size $cwnd_{abe}$. The algorithm will do nothing if *BWratio* is less than 1. This is the indicator that the network activity is increasing. In the slow start phase, once again the congested window size *cwnd* is increased by summing it with $cwnd_{abe}$. Undoubtedly, this algorithm helps a lot to increase the transmission rate of packet more quickly in a network with a large end-to-end delay such as in satellite networks.

3.2 Proposed Packet Loss Control Phase

In TCP, packet losses have occurred upon the receiving of *DUPACK*s and the occurrence of RTO. The algorithms in Fig. 2 and 3 will be activated after the detection of *DUPACK*s and RTO respectively. The purpose of these algorithms are to handle packet losses in TCP transmission by retransmitting all the packets that loss in the network. The policy of these algorithms are to prohibit the reduction in transmission rate upon the detection of packet losses that caused by bit error. In both algorithms, *ssthresh* value will be updated by the largest value between the size of $cwnd_{abe}$ and the half value of $cwnd_{prev}$. By doing so, unnecessary reduction of the *ssthresh* size can be avoided.

```
Get the size of cwnd_prev/2
Get the size of cwnd_abe
If cwnd_abe > cwnd_prev/2
        ssthresh = cwnd_abe
else
        ssthresh = cwnd_prev/2
if cwnd_prev < ssthresh
        cwnd = cwnd_prev
else if (cwnd_prev ≥ ssthresh)
        cwnd = ssthresh
```

Fig 2: After the Detection of Packet Losses Algorithm by *DUPACK*s

```
Get the size of cwnd_abe :
Get the size of cwnd_prev/2;
If cwnd_abe > cwnd_prev/2
        ssthresh = cwnd_abe;
else
        ssthresh = cwnd_prev/2;
cwnd = 1;
```

Fig 3: After the Detection of Packet Losses Algorithm by RTO

In Fig. 2 and 3, *cwnd*, $cwnd_{prev}$, $cwnd_{abe}$ represent updated congestion window, previous congestion window and estimated window size respectively.

4. Experiment Setting

There are five performance metrics which are used in this research. They are throughput, average delay, packet loss, congestion window and transfer duration. Throughput refers to the volume of data (in Mbps) that can flow through a network. It is the average rate of successful number of packets sent over a communication channel (i.e satellite links in this case) from source to the destination. The average delay specifies the average duration (in seconds) of each TCP packet takes to travel across the satellite network from source to destination. Packet loss occurs when one or more packets of data travel across a satellite network fail to reach their destination. It is measured as the percentage of dropped packets per size of bytes sent. Packet loss may also be defined as the packets that are retransmitted due to corrupted or loss. Congestion window refers to a flow control imposed by the sender. It is one of the factors that determine the number of bytes that can be outstanding at any time. Lastly, download duration refers to the time taken (in seconds) for the satellite network to transfer certain size of file (in MB) from source to destination.
For the simulation, network simulator *ns*-2 version 2.35 is used as downloaded from http://www.isi.edu/nsnam/ns/ with satellite extension module by considering LEO Iridium constellation [10][11].
Fig. 4 shows the standard satellite network topology that is implemented in the simulation which has been adopted from [11][12]. For this experiment, the source satellite which is defined as X0 (Fig. 7) is located at Serdang, Malaysia with coordinate (3.02, 101.7) latitude and longitude respectively. The location of the destination satellite which defined as X1 (Fig. 4) is at Tsetserleg, Mongolia with coordinate (47.55, 101.7) latitude and longitude respectively. According to the topology in Fig. 4, S0 is set as TCP source which carry traffic load to the destination D0 (TCP sink). S1 and S2 are UDP sources with D1 and D2 are their destinations respectively. A link between X0 and X1 is set as the satellite link, and the random packet loss occurs in this link. The bandwidths of upstream links which connect all the sources to the satellite are 1.5 Mbps as well as the bandwidths of downstream links which connect the last satellite with the destination nodes.
In this simulation, all the source and destination nodes share the bottleneck satellite link with the bandwidth set at 1.0 Mbps. The supposed TCP application in this simulation is FTP which has only 1 flow compared to UDP applications which have 2 flows. The network model is simplified with limited bandwidth size and the number of node in order to clearly show the effectiveness of the proposed algorithm under heavy congested link. All the simulation results are obtained by averaging ten trials for better accuracy.

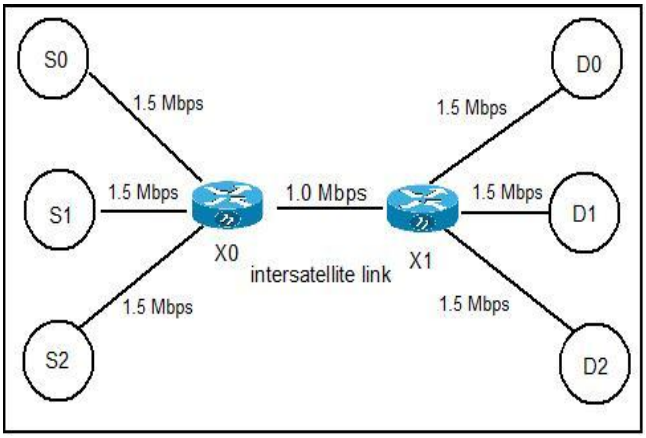

Fig 4: The Network Topology

Table 1 shows the specification of the global configuration satellite network parameters. These standard parameters are taken from [11][12] which simulates iridium satellite constellation as the simulation model.

Table 1: Global Configuration Parameters of Satellite Network

Parameter	Value
Constellation	Iridium
Altitude	780 KM
Orbital period	6026.9 sec
Inclination	86.4 degree
Downlink bandwidth	1.5 Mbps
Uplink bandwidth	1.5 Mbps
ISL bandwidth	1 Mbps
UDP rate	410Kbps

5. Results and Discussion

The experiment is conducted to compare the performance of TCP Westwood, TCP WestwoodNew and Mod TCP WestwoodNew. Note that the term Mod TCP WestwoodNew which can be identified in the result's represents the enhanced of TCP WestwoodNew which describes this research.

5.1 Throughput and Congestion Window (*Cwnd*) Analysis

This section evaluates throughput and congestion window (*cwnd*) size. The duration of this experiment is set at 60 seconds of satellite operation time. The rate of UDP applications with constant bit rate (CBR) are both 410Kbps with packet size of 510 Bytes. In this experiment, TCP application starts at 0.5 second and stop at 60th second. Both UDP1 and UDP2 application stream on the other hand, start at 2.0 and 3.0 seconds respectively and stop at 55.0 seconds. The purpose of TCP and UDP applications are not to be run simultaneously is to emphasize the impact on TCP flow in presence of UDP. UDP flows in this case, theoretically, will affect the performance of TCP and tend to monopolize the bandwidth as long as the connection is persistent. At 55 seconds, the UDP flows are stopped and the bandwidth will be released back to TCP again. The average duration of the simulation is about 8 to 10 hours.
Fig. 5 shows the throughput results of the compared TCPs. Among those TCPs, Mod WestwoodNew has the highest throughput compared to Westwood and WestwoodNew even when the bottleneck link is over congested with UDP packet

flows. When the UDP flows stop at 55.0 second, mod TCP WestwoodNew regain the bandwidth faster compared to Westwood and WestwoodNew. These are due to the fact that according to the proposed algorithm in Fig. 1, if the bandwidth ratio is greater than 1.5, the *cwnd* size is obtained by summing the window size of default TCP Reno and the absolute estimated window size $cwnd_{abe}$ which is based on the current network status as opposed to the fixed congestion window *cwnd* used in the other two algorithms. Apparently, this method can increase the TCP flow rate quickly after suffering heavy congestion. Moreover, by summing the window size of default TCP Reno with half the value of $cwnd_{abe}$ when the *BWratio* is less than 1.5 could maximize the TCP flow in the best effort manner.

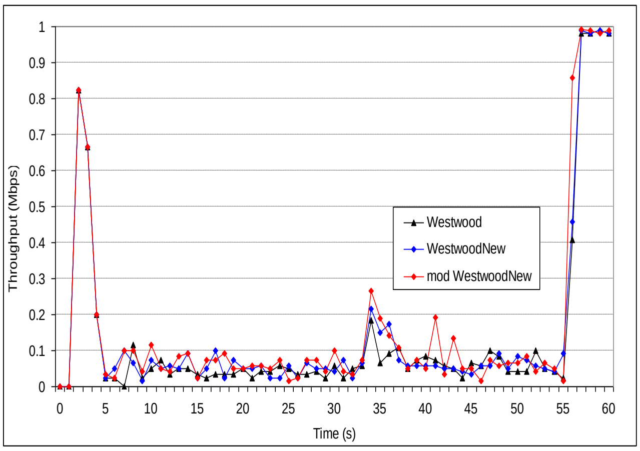

Fig 5: Throughput versus Time

Fig. 6 depicts the result of the comparison of congestion window size among the compared TCPs. For this experiment, the maximum congestion window size is set at 55. Theoretically, the throughput result as shown in Fig. 5 reflects the congestion window size result in Fig. 6. If the throughput increase, the congestion window size is also increase and vice versa. The general behavior patterns of both charts have some similarities. Both charts show that there are drastically increase and decrease in terms of throughput and congestion window size (*cwnd*) after 0.5 and 3 seconds respectively. The decrease of *cwnd* and throughput after 3 seconds are due to the presence of UDP flows which try to dominate the link. Both *cwnd* and throughput once again increase drastically to maximum value after UDP flows are disconnected.

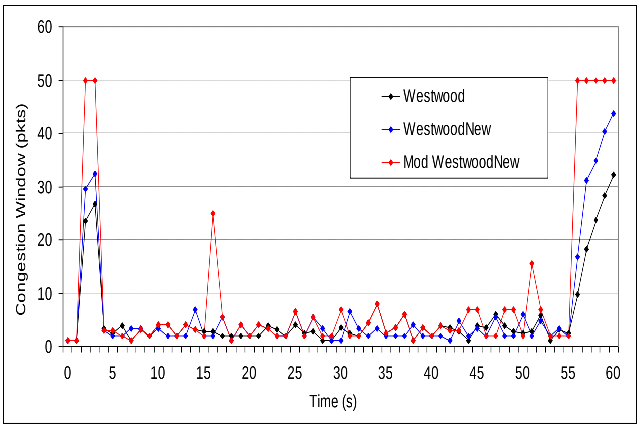

Fig 6: Congestion Window Size versus Time

Fig. 6 illustrates the overall congestion windows size of Mod TCP WestwoodNew which is bigger than that of TCP Westwood and WestwoodNew. This is due to the fact that by exploiting the precious estimated available bandwidth as described earlier from the proposed algorithm in Fig. 1 could possibly enlarge the size of congestion window and increase the sending rate optimally.

5.2 Transfer Duration Analysis

This experiment is to measure the performance of TCP Westwood, WestwoodNew and Mod WestwoodNew in terms of the transfer duration of certain size of file. The TCP with the lowest transfer duration is considered the best. The simulation of file transfer with the initial size of 1MB is considered for this experiment. The experiment will be repeated by increasing the size of file by 1MB. The experiment is stopped when the increment size of a file reaches 10MB. During the experiment, there is no time limit. The simulation will be kept running until the target size of files achieved. At the end of each repeating experiment, the duration of a file transferred is recorded. Based on Fig. 7, for transferring the size of 10MB file, Mod TCP WestwoodNew only consumed about 1600 seconds as compared to TCP Westwood and WestwoodNew which took about 1650 seconds and 1780 seconds respectively. This implies, the Mod TCP WestwoodNew requires shorter time to transfer the file compared to TCP Westwood and WestwoodNew. This is because by manipulating the estimated available bandwidth which has mentioned earlier, it could speed the increment of throughput and increase the size of congestion window. Thus, the time taken for file transferring could be significantly reduced.

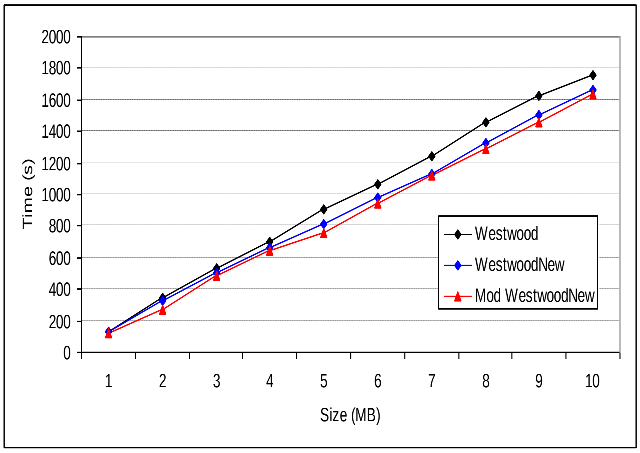

Fig 7: Time versus the Size of File

5.3 Packet Loss Analysis

Packet loss is one of the factors that can be measured to determine the efficiency of a network. In this research, the packet loss rate from the percentage of packet dropped is determined. Fig. 8 is the graph of the percentage of packet drop versus size of data in byte unit. The experiment starts with the initialization of data size at 0.5 MB. The experiment is repeated with the increment of data size by 0.5MB. The experiment stops until the data size reaches 3 MB. At the end of each experiment, the number of packet drops is recorded.

Based on Fig. 8, the percentage of packet drop is directly proportional with the size of data sent. The worst packet loss is suffered by TCP Westwood followed by Mod WestwoodNew and WestwoodNew. There is not much difference between Mod WestwoodNew and WestwoodNew in terms of packet loss rate. Packet loss rate of Mod WestwoodNew shows slightly higher than that of WestwoodNew. This is due to the fact that the tendency of Mod WestwoodNew to send more segments compared to WestwoodNew in heavy congested links in attempt to increase the throughput rate. With the insignificant difference between Mod WestwoodNew and WestwoodNew in terms of packet loss rate, it can be concluded that Mod WestwoodNew is able to increase the throughput rate while maintains the packet loss rate as WestwoodNew.

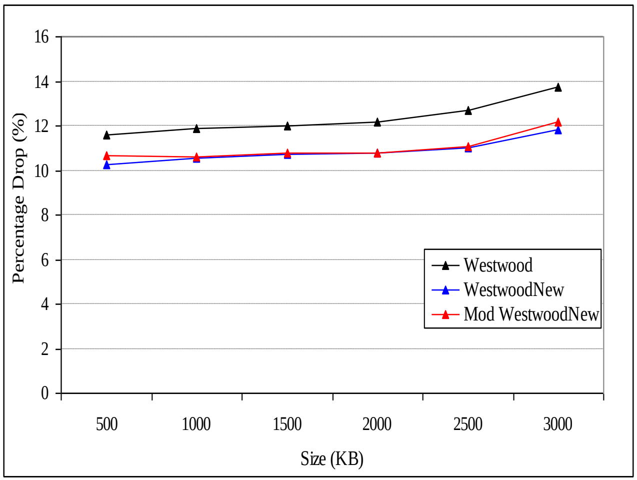

Fig 8: Percentage of the Packet Drop versus the Size of Data

5.4 Average Delay Analysis

Fig. 9 shows the comparison of the average delay of TCP Westwood, WestwoodNew and Mod WestwoodNew. Average delay is the average time (in seconds) taken for all TCP packets sent from source to destination. According to the Fig., TCP Westwood recorded the highest average delay followed by Mod WestwoodNew and WestwoodNew. Once again there is not much difference in terms of time taken for the packet sent between Mod WestwoodNew and WestwoodNew. The result shows the average delay of Mod WestwoodNew is slightly higher than that of TCP WestwoodNew. The difference is because of there are more same packets to be resent by Mod WestwoodNew due to packet loss in the heavy congested links with the presence of UDP traffic. In return, Mod WestwoodNew is able to increase its throughput rate compared to TCP WestwoodNew.

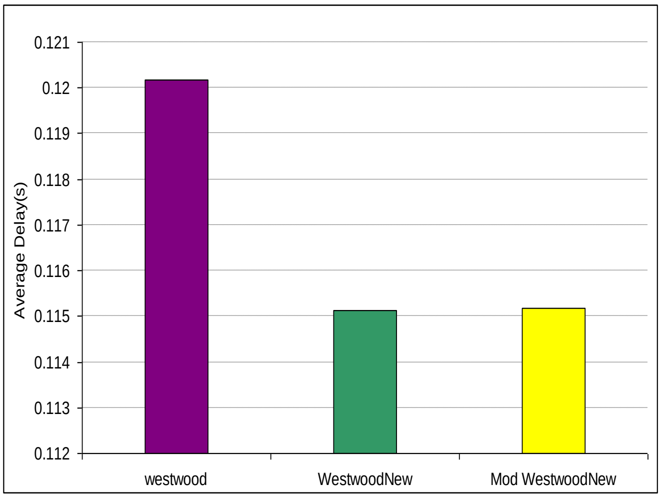

Fig 9: The Comparison of the Average Delay of the TCPs

6. Conclusion

In this research, an enhanced TCP WestwoodNew has been implemented for LEO satellite networks. The enhancement mainly focuses on the congestion avoidance and packet loss control phases of TCP WestwoodNew. A series of simulation experiments have been made to evaluate TCP Westwood, WestwoodNew and the proposed enhanced TCP WestwoodNew in terms of their performances. From the results of the performance evaluation, the proposed congestion avoidance algorithm obtains higher throughput in heavy congested satellite links with the presence of UDP traffic compared to previous TCP WestwoodNew's congestion avoidance algorithm while attempting to maintain the average delay and the packet loss rate.

References

[1] Low, S. H., Paganini, F., and Doyle, J. C., 2002, "Internet Congestion Control," *IEEE Control Systems Magazine*, Vol. 22, No. 1, pp. 28-43.

[2] Lien, Y. N., & Hsiao, H. C. (2007, June). A New TCP Congestion Control Mechanism over Wireless Ad Hoc Networks by Router-Assisted Approach. In *Distributed Computing Systems Workshops, 2007. ICDCSW'07. 27th International Conference on* (pp. 84-84). IEEE.

[3] Hagag, S., & El-Sayed, A. (2012). Enhanced TCP Westwood congestion avoidance mechanism (TCP WestwoodNew). *International Journal of Computer Application*. *45*(5).

[4] Ahmad, M., Hussain, M., Abbas, B., Aldabbas, O., Jamil, U., Ashraf, R., & Asadi, S. (2018). End-to-End Loss Based TCP Congestion Control Mechanism as a Secured Communication Technology for Smart Healthcare Enterprises. *IEEE Access*, Vol. *6*, pp. 11641-11656.

[5] Cloud, J., Leith, D., & Medard, M. (2014). Network Coded TCP (CTCP) Performance over Satellite Networks. *arXiv preprint arXiv:1310.6635*.

[6] Nguyen, T. A. N., & Sterbenz, J. P. (2017). An Implementation and Analysis of SCPS-TP in ns-3. In *Proceedings of the Workshop on ns-3* (pp. 17-23). ACM.

[7] Pirovano, A., & Garcia, F. (2013). A New Survey on Improving TCP Performances over Geostationary Satellite Link. *Network & Communication Technologies*, *2*(1).

[8] Caini, Firrincieli, Marchese, Cola, Luglio, Rosetti, & Pototi, (2007). Transport layer protocols and architectures for satellite networks. *International Journal of Satellite Communications and Networking*, *25*(1), 1-26.

[9] Casetti, C., Gerla, M., Mascolo, S., Sanadidi, M. Y., & Wang, R. (2002). TCP Westwood: End-to-end congestion control for wired/wireless networks. *Wireless Networks*, *8*(5), 467-479.

[10] Henderson, T. R., & Katz, R. H. (2000a). On distributed, geographic-based packet routing for LEO satellite networks. In *IEEE Global Telecommunications Conference 2000 GLOBECOM'00*. Vol. 2, pp. 1119-1123

[11] Henderson, T. R., & Katz, R. H. (2000b). Network simulation for LEO satellite networks. *American Institute of Aeronautics and Astronautics*.

[12] Henderson, T. R. (1999). *Networking Over Next-Generation Satellite Systems* (Doctoral dissertation, UNIVERSITY of CALIFORNIA).