

Congestion control in MANET by analyzing network status - New ETCP-Westwood

Sejal K.Haveliwala^{1*}, Dr. Dhaval Vyas², Dr. Hemal Shah³.

^{1*} M.Sc.IT department, Shree M.M.Patel Institute of Sciences and Research, Gandhinagar, India.

² Master of Philosophy, C. U. Shah University, Wadhwan, India.

³ ICT department, Ganpat University, Mehsana, India..

^{1*} sejal.haveliwala@gmail.com, ² cudeanmphil@gmail.com, ³ hemal.shah@ganpatuniversity.ac.in

Abstract:

Transmission control protocol provides congestion control mechanisms. These mechanisms work well for mobile adhoc network and wireless networks. TCP provides different mechanisms for both networks. TCP suffers from the packet loss problem in both networks. TCP controls congestion as per type of loss. The mobile adhoc network faces two types of packet loss, 1) loss due to congestion 2) loss due to link failure. TCP can not identify these loss types and it assumes loss as congestion loss. It has existing variants which perform congestion control. One of TCP variant TCP westwood works for congestion control with the help of bandwidth estimation. It works well but has not taken care the value of last Round trip time and estimated round trip time which are the important parameters to analyze network status. The proposed westwood has taken care of these two parameters and controls congestion window based on them. The work proposed in this paper focuses utilization of round trip time for throughput improvement. The proposed algorithm is tested on NS 2.35 and compared with existing variants like TCP Reno, TCP Vegas and TCP westwood and result reveals certain improvements.

Keywords: MANET (mobile adhoc network), cwnd (congestion window), ssthresh (slow start threshold), RTT (round trip time), RTO (retransmission time out), DCM (dynamic congestion control for MANET), ERTT (estimated round trip time)

1. INTRODUCTION

TCP has its tremendous role for today's Internet connectivity. Due to the rising need for communication, a different type of networks like wireless and mobile adhoc network are introduced. TCP variant supports mobile adhoc networks. It works well for MANET, but TCP has challenges in loss differentiation and loss recovery [1][2][4-6]. TCP does not face these challenges in a wired network. MANET supports communication between independent wireless nodes without any wired infrastructure. TCP faces challenges to differentiate loss whether it is congestion or link failure [7]. It always interprets that loss is due to congestion and reduce congestion window. This degrades the performance. Loss due to link failure may also occur in MANET because of mobility or battery usage. This loss can be identified when retransmission time out occurs. Haveliwala et al., 2020 have observed that during link failure if bandwidth estimation is done properly, the performance can be improved [8]. One of TCP's variant TCP Westwood works on this mechanism and controls congestion window as per bandwidth estimation. In MANET, the Round trip time is also an important parameter to control congestion. It is used to identify network status, whether it is congested or not congested. With the help of this status congestion window is set. The proposed westwood has been implemented with this modification and hence it gives remarkable throughput compared with the other TCP variants like, Reno, Westwood, WestwoodNR. Here we studied the different variants of westwood discussed in section -2 and came to know that all authors have worked on the same problems using different mechanism. Section 3 has discussed about proposed algorithm, section 4 shows performance evaluation and section 5 concludes the paper.

2. RELATED WORK

TCP faces the problem of congestion control during a link failure in MANET. Many researchers have worked on this problem to enhance TCP's performance using bandwidth estimation with variant TCP Westwood. They have applied different ideas on this and gained improvement compared with other variants. We surveyed the research done on the same problem to get better performance.

2.1 TCPW

Authors Casettiet al., 2002 have proposed the algorithm which is a modification of TCP Reno [3]. The new name given is TCP Westwood (TCPW). The algorithm worked on window control. It also worked on 3 DUPACK. The values of CWND and ssthresh are set with the help of bandwidth estimation. They suggested that Reno is halving the cwnd without bandwidth estimation. The new approach avoids this. It gives fast recovery and congestion avoidance. Algorithm considered the case of before and after congestion. After congestion, bandwidth is calculated, which is used to set cwnd and ssthresh. Before congestion, as per available bandwidth the cwnd is adjusted rather than halving it unnecessarily. The main aim is to set cwnd properly and that can work on both conditions, after n DUPACK or after timeout occurs. They have found the results in terms of better throughput, delay performance, end to end propagation times and goodput. They have improved result because it avoids unnecessary reduction in cwnd and ssthresh which is not dependent on RTT. They received higher values in cwnd and ssthresh which leads high throughput.

2.2. Proportional study of TCPW and other variants

Authors Razaque and Elleithy, 2014 have analyzed issues of MANET with mobility and examines the impact on other TCP variants. They have found that TCP westwood can give better performance. They have worked with a different movement of nodes, speed and Random waypoint mobility model. They have shown the fairness of RTT and identified that increased value of RTT mean it uses more resources and this gives degradation in throughput. They have also analyzed that with the increasing speed TCP's performance degrades [9].

2.3. TCP BRJ

In TCP BRJ, authors Mu et al., 2015 have proposed bandwidth estimation in slow start phase, which divides congestion based on RTT jitter in congestion avoidance. It also differentiates congestion and random packet loss and works as per type of loss. When acknowledgement of a new data packet has been received, their algorithm calculates RTT jitter and classify the network into five different classes of congestion. TCP BRJ also differentiated random packet loss from congestion loss when three duplicate acknowledgments are received and works accordingly. They found remarkable improvement in throughput, bandwidth estimation and fairness compared in other networks with high bit error rate. BRJ works for the heterogeneous network [10].

2.4. TCP DCM

Hamamreh, 2019 has given the new approach for secure and dynamic path congestion estimation. They have studied the other variants like TCP new reno, TCP vegas, TCP westwood and TCP welcome. They have analyzed how these variants work for loss differentiation and recovery in MANET. They have also worked on TCP's inability to judge the type of loss in MANET. They have proposed, TCP DCM, Dynamic end to end congestion detection protocol for MANET. The proposed algorithm works in two phases ,1) Destination selects three valid routes with minimum cost and send reply on them. 2) The source finds the minimum RTT path and selects it for next communication. The

source also selects new cwnd for it. They have also focused on the selection of cwnd as selection of cwnd is high then the network may face congestion and if low value is selected network performs with low throughput. The average queue length has been used to find the network's congestion status. The better performance in congestion control has been found and compared with other variants like TCP new reno and TCP welcome because dynamic path selection was used here. Whereas others have not used it [11].

2.5. DCM+

Hamamreh and Khader, 2019 have proposed above mechanism TCP DCM and have extended the same work with name DCM+. Here they have used bandwidth estimation which was used in westwood+. The new idea is a combination of New reno and westwood+. It has updated congestion avoidance mechanism. It has used Bandwidth estimation and calculated the ssthresh and cwnd. This calculation has also taken care of RTT, RTO and a rate of congestion avoidance. This rate is used to calculate next cwnd. DCM+ can work with four phases i) slow start ii) congestion avoidance iii) fast retransmission and iv) fast recovery. Because of its dynamic behavior, it has performed well and given improved throughput, low delay and a smaller number of packet loss [12].

3. PROPOSED ALGORITHM

All surveyed algorithms work on the problem of congestion control using different ideas. All authors work on bandwidth estimation to set congestion window. TCPW has worked for congestion control in two cases ,1) when 3 dupack are received and 2) when time out occurs as suggested by Casetti et al., 2002[3]. They have taken care of unnecessarily reduction in cwnd and ssthresh , which was not considered previously and hence they found improvement in throughput as compared with others. Their mechanism doesn't depend on RTT value. While with proportional study of TCP's different variants, authors Razaque and Elleithy, 2014 have studied about the fairness of RTT [9]. They identified that increased value in RTT means, more resources are utilized and hence it causes down performance. TCP BRJ also used RTT jitter to classify network congestion and got improved performance. TCP DCM also works with analysis on RTT's value. Here Hamamreh, 2019 has shown the mechanism with a dynamic path selection which has a minimum value of RTT and hence TCP DCM achieved better throughput than other[11]. DCM+ is an extension of DCM, here Hamamreh and Khader, 2019 have taken care RTT, RTO and congestion avoidance rate to set cwnd and ssthresh. Hence DCM+ performs well as compared with DCM[12]. All above researchers have worked to improve the network performance by controlling congestion. Most of them used RTT for betterment. They have not worked when link failure occurs. When Retransmission time out occurs TCP interprets as congestion loss and executes it's congestion control algorithm, this degrades performance. Hence Khelage, 2014 suggested that TCP fails here when link failure occurs. During this time period cwnd must be controlled differently. It is done with taken care of network status about congestion. Researchers have not controlled cwnd when RTO occurs. TCP new westwood has mechanism which manages cwnd differently when RTO occurs. As per the above-mentioned researcher's work we have analysed the RTT to set cwnd.

Our algorithm has checked the last and current value of RTT and state the congestion status of network. Route is found congested when last RTT is greater than estimated RTT. Algorithm sets the value of cwnd when the route is found congested. The cwnd and ssthresh is also compared before setting new cwnd. It sets cwnd with the addition of higher value as per the ratio of lastRTT and ERTT, when cwnd is less then ssthresh and when it is greater than ssthresh new cwnd is set with addition of lower value .Hence the algorithm controls congestion during link failure and provides better bandwidth utilization. Because of this control packet loss is be recovered and performance is increased.

```

if (last_rtt > ERTT){
printf(" route is congested");
if (cwnd_ < ssthresh_)
    cwnd_ = cwnd_ + ((last_rtt / ERTT) * 0.9);
if (cwnd_ > ssthresh_)
    cwnd_ = cwnd_ + (ERTT / (cwnd_ * last_rtt));
}

```

Here,

$$\text{ERTT} = t_rtt_ * \text{tcp_tick_}; \quad (1)$$

tcp_tick is simulation time unit in seconds, default it is 100 ms, 0.1 sec.

$$\text{Last_rtt} = \text{Scheduler::instance().clock()} - \text{ts_peer_}; \quad (2)$$

(ts_peer is the latest Timestamp provided by peer TCP receiver.)

Both (1) and (2) shows how ERTT and Last_rtt is calculated respectively.

4. PERFORMANCE EVALUATION

The performance of the algorithm is evaluated in terms of throughput and delay, calculated from the executed simulation scenario.

i) Performance Metrics

(1) Throughput, is defined as the performance of network in terms of out of sent number of packets how many are received by receiver within specified time. $\text{Throughput}(\text{bits/sec}) = (\text{Delivered packets} * \text{packet size} * 8) / \text{Simulation time period}$. (2) Delay, is the measured in seconds as time period required for send data packets from sender to receiver.

ii) Simulator for simulation

With the help of simulation we can conduct testing for the new algorithms, we can also check how it will work for real scenario and how much accuracy it will give with given parameters [14].

iii) Simulation Tool:

Here we have used NS2.35 for our implementation and experimental evaluation. NS2 is written in C++ which is object oriented language. It supports implementation of different TCP variants and also provides the platform to test them by generation different scenarios for wired, wireless and MANET.

We have taken the dimension of topography X and Y are 500X500 and node density is of 30 nodes. Nodes are distributed using random waypoint mobility model. We generated the communication between nodes 0 and node 12. Velocity for the movement of nodes is taken random. The packet size here is kept 1000 bytes, by default as TCP has taken. Bandwidth taken here for mobile node is 10 Mbps. Simulation is performed up to 50 seconds.

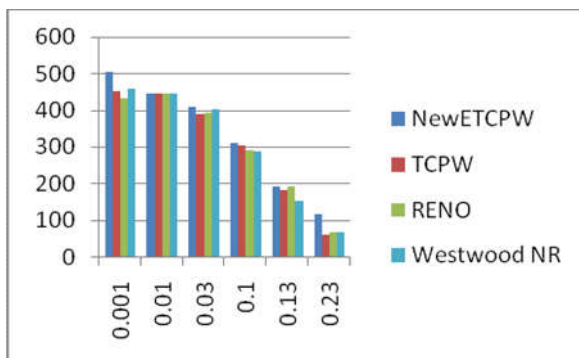
The proposed algorithm works for link failure condition and it is tested for that. To generate link failure Uniform error model is used with the TCL file when we generated the scenario. The algorithm is tested for different error rates.

iv) Performance Evaluation:

Table 1. Throughput comparison of new TCP westwood and other TCP variants

Error rate	Throughput (Kbps)			
	newETCPW	TCPW	TCP Reno	TCP WestwoodNR
0.001	503.397	452.034	431.240	457.463
0.01	444.804	445.121	444.065	444.065
0.03	408.813	389.311	392.661	402.467
0.1	308.56	304.837	288.992	286.307
0.13	191.842	181.38	192.542	151.179
0.23	115.427	59.074	64.919	65.809

The table 1 shows the throughput comparison of proposed algorithm (named as New Enhanced TCPwestwood) with other specified TCP variants. As the packet transfers from source to destination ,because of error generation the retransmission time out occurs and at that time the condition for network congestion status is checked with the help of RTT's value and then congestion window is controlled dynamically if route is congested or vice versa. As a result, we recorded the trace file generated by the execution. With the help of AWK script we calculated the throughput for each mentioned TCP variants. Table suggested that the algorithm works properly and controls cwnd as per network status, hence we received good throughput as compared with others. Figure 1 shows the comparison chart. It suggests that new TCP westwood performs well when there are more link errors. At higher error rate 0.23 the new TCP westwood gives highest throughput 115.42kbps as compared with others.

**Figure 1. Throughput comparison of new ETCP westwood and other TCP variants with different error rates.****Table 2. Delay comparison of new ETCP westwood and other TCP variants**

Error rate	Delay(seconds)			
	newETCPW	TCPW	TCP Reno	TCP WestwoodNR
0.001	0.1657	0.1968	0.1865	0.1955
0.01	0.1966	0.1968	0.2042	0.1884
0.03	0.2156	0.2356	0.2378	0.2268
0.1	0.2505	0.2409	0.2370	0.2370

0.13	0.1976	0.1597	0.2097	0.2936
0.23	0.2115	0.1367	0.1461	0.2411

Like throughput, we have also compared delay and table 2 shows that. It is observed that we got reduced delay for new Westwood. Figure 2 shows the comparison.

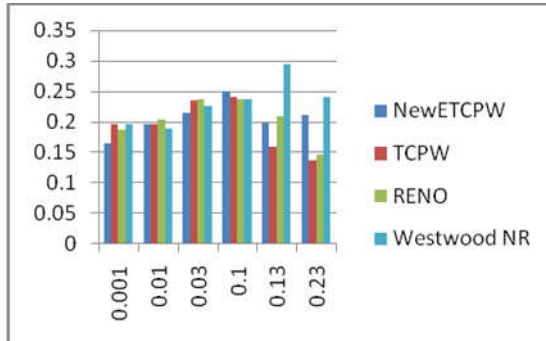


Figure 2. Delay comparison of new ETCP westwood and other TCP variants with different error rates.

5. CONCLUSION

The proposed algorithm solves the problem of TCP for MANET to control congestion during link failure. The presented TCP variant Westwood has improved performance in MANET. Failure has been identified by the retransmission time out period and we set congestion window as per network status about congestion. Network status has been identified by the RTT s. The proposed algorithm has been compared with the other TCP variants like westwood, Reno and westwoodNR. It handles the failure in more effective manner than other variants, hence it improves the throughput for MANET.

6. REFERENCES

- [1] Amin, M., Fard, K., Karamizadeh, S., & Aflaki, M. (2011). Enhancing Congestion Control To Address Link Failure Loss Over Mobile Ad-Hoc Network. *International Journal of Computer Networks & Communications (IJCNC)*, 3(5), 177–192. <https://doi.org/10.5121/ijcnc.2011.3513>
- [2] Bregni, S., & Caratti, D. (2003). Enhanced loss differentiation algorithms for use in TCP sources over heterogeneous wireless networks. ..., 00(C), 2–6.
- [3] 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. <https://doi.org/10.1023/A:1016590112381>
- [4] Fard, M. A. K., Karamizadeh, S., & Aflaki, M. (2011). Packet loss differentiation of TCP over mobile ad hoc network using queue usage estimation. *2011 IEEE 3rd International Conference on Communication Software and Networks, ICCSN 2011*, 81–85. <https://doi.org/10.1109/ICCSN.2011.6013667>
- [5] Park, M. Y., Chung, S. H., & Sreekumari, P. (2009). Estimating rate of queue usage to differentiate cause of packet loss in multi-hop networks wireless. *Proceedings - International Computer Software and Applications Conference*, 1, 500–508. <https://doi.org/10.1109/COMPSAC.2009.73>

- [6] Ye, J., Wang, J., & Liu, H. (2006). *An adaptive loss differentiation algorithm based on queue management. International Conference on Communication Technology Proceedings, ICCT, (60673164), 2–5.* <https://doi.org/10.1109/ICCT.2006.341917>
- [7] Haveliwala, S. (2018). *The Survey of Packet Loss Differentiation Techniques in TCP.* 11(2), 25–36.
- [8] Haveliwala, S., Vyas, D., & Shah, H. (2020). *Dynamic Congestion Control Mechanism in Mobile Adhoc Network: TCP Westwood-DCC. International Journal of Recent Technology and Engineering, 8(5), 4784–4789.* <https://doi.org/10.35940/ijrte.E6828.018520>
- [9] Razaque, A., & Elleithy, K. (2014). *Proportional Study of TCP Variants over Heterogeneous Wireless Networks. International Journal of Computer Applications, 107(21), 24–31.* <https://doi.org/10.5120/19146-0352>
- [10] Mu, J., Liang, Q., Wang, W., Zhang, B., & Pi, Y. (2015). *The proceedings of the third international conference on communications, signal processing, and systems. Lecture Notes in Electrical Engineering, 322, 623–632.* <https://doi.org/10.1007/978-3-319-08991-1>
- [11] Hamamreh, R. A. (2019). *SDCM: SECURE DYNAMIC END-TO-END CONGESTION AVOIDANCE PROTOCOL FOR MANETS.* 97(21), 3122–3131.
- [12] Hamamreh, R. A., & Khader, D. (2019). *DCM+: A multi-purpose protocol for congestion control. IEEE 7th Palestinian International Conference on Electrical and Computer Engineering, PICECE 2019, (July), 1–7.* <https://doi.org/10.1109/PICECE.2019.8747237>
- [13] Khelage, P. B. (2014). *TCP- Costco Reno : New Variant by Improving Bandwidth Estimation to adapt over MANETs.* 5(2), 67–76.
- [14] Mallapur, S., & Patil, S. R. (2012). *Survey on Simulation Tools for Mobile Ad-Hoc Networks. International Journal of Computer Networks and Wireless Communications, 2(2), 241–248.*