International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-5, January 2020

Dynamic Congestion Control Mechanism in Mobile Adhoc Network: TCP Westwood-DCC

Sejal Haveliwala, Dhaval Vyas, Hemal Shah

Abstract: ***Transmission control protocol faces a problem of packet loss differentiation in the wireless and mobile adhoc network. Congestion control is not properly done here. It cannot manage the congestion window as per type of loss and it reduces Congestion window unnecessarily and that degrades the performance. TCP Westwood cannot identify congestion or link failure loss, and it cannot manage the congestion window as per available bandwidth. This paper discusses that TCP Westwood performs bandwidth estimation, setting up a congestion window and a slow start threshold. In mobile adhoc network, link failure may happen frequently, and it should be handled properly. Link failure can be detected with the help of retransmission timeout. Once timeout occurs Westwood performs congestion avoidance. Proposed Westwood manages three states of congestion 1) Avoidance 2) congestion and 3) No congestion, it updates congestion window and slow start threshold as per the status of network. It maintains congestion window dynamically. Network status is identified by estimated bandwidth and proportionality ratio. Proposed method is tested on NS2.35 and compared with the existing TCP variants. The proposed Westwood performs optimized link utilization and congestion control mechanism. Hence it gives significant performance for loss recovery.***

Keywords: BWP (bandwidth proportionality ratio, Congestion window(cwnd), Estimated Round trip time (ERTT), Last RTT Round trip time (RTT), ssthresh (slow start threshold), Estimated bandwidth(BWE).

I. INTRODUCTION

Wireless network or MANET faces the problem of packet loss due to congestion, link failure or wireless Chanel error [1]-[6]. TCP is a connection oriented and reliable protocol, it works on principles of acknowledgment, but it can't differentiate packet loss properly [2]-[4],[7], [8]. TCP should be designed in such a way that it can detect a proper type of loss. TCP controls congestion by algorithms. Because of insufficient bandwidth and increased network traffic than the capacity congestion may arise. TCP runs two states 1) slow start and 2) congestion avoidance [9]. In the slow start the congestion window is incremented till it reaches to defined threshold and after that it moves into avoidance, in which TCP makes cwnd to half .It is observed that in wireless or MANET, loss can occur due to channel error and TCP reduces cwnd by assuming the loss as congestion [1], [10]. To avoid congestion TCP runs algorithms like slow start, fast retransmit, fast recover and additive increase and multiplicative decrease [11]. TCP cannot control congestion properly and can’t recognize the loss due to path disturbance or channel error. TCP introduced many variants to work on this.

The other problem of TCP is bandwidth estimation in a congestion control scheme [12]. TCP has to estimate it properly during congestion or congestion avoidance phase to set cwnd and ssthresh [13]. TCP introduced different variants like TCP Tahoe, Reno, New Reno, Sack and Westwood for congestion control. In Tahoe, no measurement of bandwidth is done. It identifies three duplicate acknowledgment as packet loss and performs fast retransmit. Reno also reduces congestion window without any loss reason consideration [11]. This causes poor performance. It cannot differ random and congestion loss[14].Reno works well in case of single packet loss and cannot manage multiple packet loss while New Reno can work with multiple packet loss[15].New Reno does not wait for 3 duplicate acknowledgment before retransmission of a lost packet ,Reno and New Reno halves the ssthreshold and reset the cwnd to one. New Reno also uses fast recovery and fast retransmission to solve the problem of Reno[10].In Sack a receiver acknowledges out of order packets selectively rather than sending all received packets’ acknowledgments .This way sender identifies missing packet and retransmit that only rather than all unacknowledged packets. Sack also gives improved performance in case of multiple packet loss [15]. Westwood utilizes bandwidth in batter way and behaves stable as compare with other, hence it gives improved performance. TCP Westwood does not half cwnd like TCP, but it manages cwnd and ssthresh as per available bandwidth. It is a rate-based algorithm [16]. It measures available bandwidth and uses network resources effectively. The modification for dynamic congestion control as per bandwidth proportionality ratio and network status regarding congested, not congested and congestion avoidance is proposed in this paper. The performance of proposed work is compared with TCP variants like Reno, Westwood and WestwoodNR.

Overview of other TCP Westwood versions related to this work is in the next section and section III presents proposed algorithm. The result evaluation, the conclusion and the future improvement are given in section IV.

Manuscript published on January 30, 2020.
* Correspondence Author

Sejal Haveliwala *, M.Sc.IT Department, Shree M. M. Patel Institute of Sciences and Research, Gandhinagar, India. Email: sejal.haveliwala@gmail .com.

Dr. Dhaval Vyas, Master of Philosophy, C. U. Shah University, Wadhwan, India. Email: cudeanmphil@gmail.com.

Dr. Hemal Shah, ICT Department, Ganpat University, Mehsana, India. Email: hemal.shah@ganpatuniversity.ac.in

II. BACKGROUND

Many researchers have worked for this area to improve TCP's variants with the help of good congestion control and avoidance mechanisms.

All variants used different congestion control algorithms, and they differ in working when link failure loss arise. They provided different approaches to improve TCPW and recover the loss.

A. Westwood NARAS

This scheme gives congestion controlling using the estimation of bandwidth and it avoids unnecessary reduction in cwnd and ssthresh. When 3 duplicate ACKs are received, it means that network is loaded upto its capacity. Here bandwidth is estimated as BW=BWE * RTT. In a wireless network TCP Westwood performs congestion avoidance in slow start phase and increases cwnd with decreased rate where it is required, and bandwidth is available. So, Westwood utilizes bandwidth in an efficient manner and gives enhanced performance [9].

B. Enhanced TCP Westwood with slow start phase:

New slow start method is defined to set ssthreshold to use bandwidth estimation. To consider link utilization the link capacity is calculated by (1),

ELC (Estimated link capacity) = $\frac{Acked * segmentsize}{tk - tk-1}$ (1)

The algorithm uses moving average method to update ELC by every Ack using (2),

$$ELC = (1-\alpha)ELCi + \alpha * ELCi - 1 \quad (2)$$

α =0.9 then ssthresh is calculated by,

ssthresh=$ELC * MinRTT$, where RTT is minimum round trip time, here ssthresh is calculated according to link capacity. Algorithm also checks the values of cwnd and last RTT in slow start phase. It sets cwnd in this said state. Hence modification improves throughput of TCPW [7].

C. TCPW with Bulk Repeat:

The bulk repeat has three modifications 1) Bulk retransmission: retransmit unacknowledged packet immediately when loss is detected.2) Fixed timeout when consecutive loss occurs.3) Window adjustment: cwnd is kept fixed in case of loss. TCPW (BR) has good performance in case of more loss. The bulk retransmission raises the questions like, how many time senders retransmit in a window? Is the window be advanced for lost packets? To solve these questions the algorithm TCPW is proposed. It performs fast recovery when a sender receives three duplicate acknowledgements or retransmission time out occurs. It retransmits unacknowledged packets within a congestion window and lost packets when there is no congestion. It sets RTO for fixed retransmission timeout. Here cwnd and ssthresh is set when loss is detected. The cwnd is not reduced when it is larger than ssthresh. So, more packets can be transmitted when RTT occurs because of the large cwnd. Rather than reducing cwnd it sets is as ssthresh when it will be greater than ssthreh when loss is detected, and it is not of congestion loss. Hence it performs with improved throughput [18].

Westwood performs fast recovery by resetting congestion window to 1 after a timeout, TCP Reno also performs fast recovery in the same way. This is a fair option because it does not take full advantage of the BWE information, it also avoids reduction of the congestion window upto 1 in the presence of losses due to interference in wireless link, rather than to congestion [19].

D. TCP Westwood New:

Enhanced TCP congestion avoidance mechanism uses data receiving rates as the important parameter to predict network condition. It estimates current bandwidth and finds the ratio between current and previous bandwidth. If ratio less than 1 means, there is increase in network load and if ratio greater than 1 there is decrease in network load. Here cwnd is set as per the network condition. This works for efficient congestion avoidance. It performs the same until timeout occurs or three duplicate acknowledgements were received. Here RTO calculation algorithm is also modified. RTO expiration can also indicate congestion in network. During congestion state TCP starts slow start. If RTO occurs and the network has still capacity this false slow start and it may degrade the performance. In case of link failure RTO is modified by New RTO as shown in (3):

$$RTOnew = (RTT\ new / RTT\ old) RTOold.RTO \quad (3)$$

This helps to recover the link loss. TCPwestwood New performs well with improved throughput and reduced delay [15].

E. TCP Westwood NRBWP: (New Reno with Bandwidth Proportionality Ratio)

The bulk repeat uses TCP WestwoodNR, which is a rate-based technique [18]. When link failure occurs, bandwidth is always used to recover it. TCPwestwoodNR enters in congestion avoidance case. The loss occurs in a network can be either congestion loss or due to link failure. The problem with TCP WestwoodNR is differentiation of loss that when congestion or random loss (link failure) occurs. As a solution of this problem WestwoodNRBWP enhanced Transmission Control Protocol was developed [20]. It can differentiate the loss using Loss Discrimination scheme. This scheme uses the last three values of RTT. After loss identification, in congestion avoidance state, cwnd is set as per network status by TCP WestwoodNRBWP. A new bandwidth estimation method is proposed. When three duplicate acknowledgment is received, or RTO happens the new cwnd and ssthresh is set accordingly. The rate of received ack is also used for calculation.

If the loss is identified because of link failure, there should no deduction in cwnd. According to network status the cwnd should be calculated. In case of link failure, a network is able to send data with a higher rate so cwnd can be increased. Here in TCPWestwoodNRBWP case the cwnd can be calculated based on BWP.BWP is a ratio between current and previous bandwidth.

F. Enhanced Westwood:

TCP Petra has been proposed which estimates the bandwidth when timeout happens, and packets are dropped. Here modified timeout procedure is defined to achieve better performance than TCPW [21]. They first check the values of the estimated bandwidth along with last RTT before changing ssthresh and cwnd. The algorithm sets cwnd for three states 1) congestion 2) no congestion and 3) congestion avoidance.

Here the key idea is that once the timeout expires, the current link status is checked by two indicators.1) estimated bandwidth 2) last RTT.

If packet loss and cwnd< EBW/2 then it indicates no congestion. If last RTT < estimated RTT, it means link is in good condition, we can increment cwnd instead of reducing it by one segment. They got the improved bit rate then TCPW.

III. PROPOSED ALGORITHM

All mentioned algorithms have worked on the bandwidth estimation for congestion control. Authors examined various techniques. They found congestion window should be changed as per a network status. Westwood NARAS has worked to identify the status of a network with the help of the reception of three duplicate acknowledgements and it also performs congestion avoidance with the help of bandwidth estimation [9]. Enhanced Westwood works with a slow start phase [17]. It has taken care of link capacity estimation. According to the estimated capacity it calculates the slowstart threshold. This algorithm works well in a slow start phase. Westwood with bulk repeat have set ssthresh and cwnd when loss is detected during bulk transmission [18]. This increases the throughput and performs fast recovery also. Westwood New also performs congestion avoidance as per network status, here the status is identified by bandwidth ratio [15]. They also calculated RTO, which helps to recover the loss and gives better throughput. WestwoodNRBWP [20] has also worked for the same as Westwood New [15] but here they worked for westwood NR and set cwnd and ssthresh as per network capacity and got good throughput. Enhanced Westwood is proposed as TCP petra, which estimates bandwidth as per current link status [21]. This algorithm also uses last_RTT to set ssthresh and cwnd.

All the above algorithms have worked to check network status or link capacity then estimates bandwidth. The proposed algorithm includes both parameters BWP and last_RTT to check network status and set ssthresh and cwnd. All mentioned algorithm has worked on congestion avoidance only they don't have estimation when congestion occurs or there is no congestion state. Our algorithm has worked for all these three states as per link status, so we found better throughput.

TCPWestwood DCC -with Petra and BWP- (proposed algorithm)

The algorithm works for a link failure case and bandwidth estimation is done as per network status in terms of last RTT and BWP. The result may have the increased number of successful transmissions as compared with other TCP variants. Here we proposed algorithm to handle link failure loss with improvement in TCP Westwood.

Cwnd is the one of the important parameters that amount of data to be transmitted during the time out period. It also decides whether the link is in congestion state or not and what to do for avoidance. Current and previous bandwidth estimation are important parameters to calculate a bandwidth proportionality ratio. The estimated bandwidth is used to calculate ssthresh which is generally used to decide when to avoid the congestion. Last_RTT and ERTT are used to decide whether the link is in good condition or not.

The key idea is once the timeout expires, link status can be identified by,1) estimated bandwidth ,2) last RTT and 3) BWP.TCP Petra has not used ssthresh for congestion avoidance. The proposed algorithm checks both condition cwnd< EBW/2 and BWP >1, both conditions indicate that the link is in good condition, there is no congestion. It also checks the value of estimated bandwidth and last RTT to check network status for a necessary congestion avoidance procedure. It checks BWP, if BWP < 1 which shows that channel is not capable to send more packets means congestion occurs and it has to maintain cwnd otherwise it sets cwnd to one. Proposed algorithm works for better bandwidth estimation during link failure and improves throughput as compared with other TCP variants like Reno, Westwood and Westwood NR.

The importance of used input parameters: -

Seg_size- length of the payload in bits.

BWP –It is a ratio of current bandwidth and previous bandwidth. Capacity of channel can be checked using this ratio.

If BWP<1, means a channel is not capable to send more data packets. If BWP>1, means channel is capable to send packets. Estimated bandwidth along with last_RTT is also used to check network status.

Last_RTT< ERTT means link is in good condition and we can increment cwnd instead of resetting it to one. Three states of congestion are identified by the BWP, Last_RTT and E_RTT.

E_RTT is calculated by (4)

$$E\_RTT(rtt\_estimate) = t\_rtt * tcp\_tick\_ \quad (4)$$

Here t_rtt is No. of ticks occurred so far starting from tick before the packet was sent, and tcp_tick_ is tick duration.

Algorithm:- TCP-westwood-DCC

1. Set input parameters like last_bwe_sample, current_bw, min_rtt_estimate, seg_size, BWE.
2. Check for retransmission timeout. (it is because of link failure)
3. If RTO occurs then calculate ssthresh as per available bandwidth using (5) and calculate BWP as (6).

$$ssthresh = (BWE * RTTmin)/seg \quad (5)$$

$$BWP = curr_{BW}/prev_{BW} \quad (6)$$

4. Check network status and apply congestion control mechanism as (7), (8) and (9). To check status cwnd, BWP, last_RTT, E_RTT is used here.

i) Check No congestion state using EBW & BWP. Set cwnd $cwnd = EBW/2$ (7)

ii) Congestion avoidance state. Set cwnd as ssthresh

$$cwnd = ssthresh \quad (8)$$

iii)Congestion state if BWP < 1, set cwnd

$$cwnd = cwnd + 1/cwnd \quad (9)$$

IV. PERFORMANCE EVALUATION

As the algorithm performs for three conditions congestion, congestion avoidance and no congestion, it gives improved performance in terms of throughput and delay.

i) Evaluation Parameters:

The evaluation parameters are:(1) Throughput, It is defined as the performance of network in terms of out of sent number of packets how many are received by receiver.(2) Delay, is the measured in seconds as time period required for communication or to transmit packet.(3) Packet delivery ratio is ratio of all received data packets from all destinations over the number of data packets sent by the senders. It is calculated by(10).

$$PacketDeliveryRatio = TotalDataPackets\ Received / Total\ DataPacketsSent \quad (10)$$

ii)Simulation setup:

Experiments were conducted in Ubuntu 14.04 using NS2.35 simulator. As per the Tcl file the scenario is generated shown in Fig.1. Here source node is 0 and destination node is 12. Source node sends data packets to destination via intermediate nodes. The error model contains both the mechanism and policy for dropping packets. Uniform error model is used here. Bandwidth is set 10 Mbps. The propagation delay is taken 10ms.The flow of traffic will run for 50 seconds of simulation time. Table-I shows the other set simulation parameters.

iii) Performance Analysis:

The simulation scenario has been executed for four congestion control algorithms. Enhanced TCPW Westwood (ETCPW) compared with TCP Westwood (TCPW), WestwoodNR (Westwood New Reno) and TCP Reno. The comparison has been done with throughput, delay and PDR. The throughput is compared with different error rate 0.03,0.13 and 0.23. Table -II shows the comparison and Fig.2 shows the comparison chart.

Table-I Simulation parameters

Parameter	*Value*
SIMULATOR	Network Simulator 2
MOBILITY MODEL	Random Waypoint
TRANSPORT AGENT	TCP-Westwood-DCC
APPLICATION AGENT	FTP
PROPAGATION DELAY	10 ms
BANDWIDTH	10 Mbps
ERROR MODEL	Uniform Error Model
AREA	500m x 500m
COMMUNICATION RANGE	250m
INTERFACE TYPE	Phy/WirelessPhy
MAC TYPE	IEEE 802.11
QUEUE TYPE	Droptail/Priority Queue
QUEUE LENGTH	50 Packets
ANTENNA TYPE	Omni Antenna
PROPAGATION TYPE	TwoRayGround
ROUTING PROTOCOL	AODV
SIMULATION TIME	50 seconds

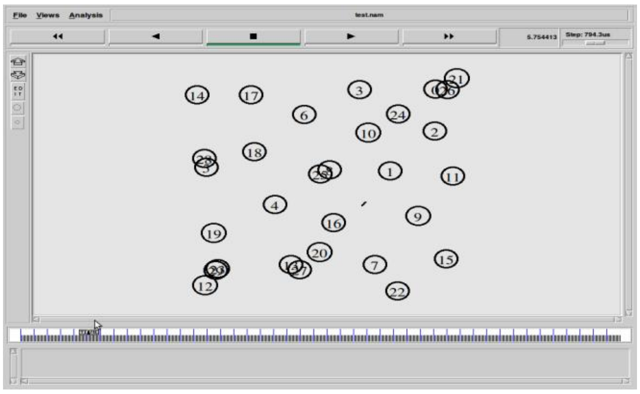

Fig.1.Nam setup in NS2

Table-II Throughput comparison with different error rates

Error rate	Average Throughput (Kbps)			
	ETCPW	*TCPW*	*TCP Reno*	*TCP WestwoodNR*
0.03	400.376	389.711	392.661	402.467
0.13	205.213	181.38	192.542	151.179
0.23	101.265	59.0744	64.9192	65.8092

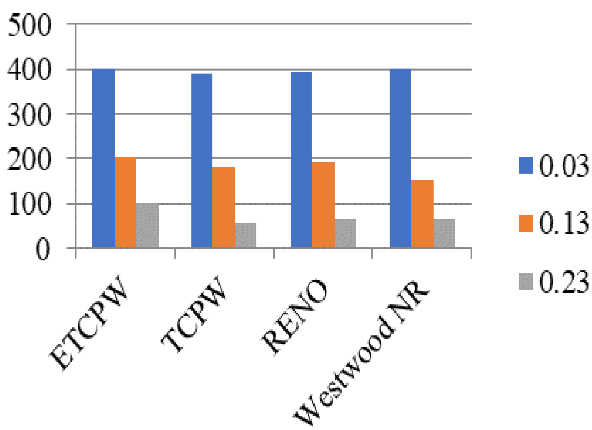

Fig.2.Throughput of different algorithms with the variation of error rate

The Table-II shows that as we increase the error rate more failure occurs the TCP Westwood DCC controls the cwnd dynamically and recover the loss, so we get the good throughput as compared with the other algorithms. The simulation is performed for 50 seconds, fig.3 shows the graph for comparison of throughput at different second for all algorithms, here error rate is 0.13. It shows that Westwood DCC performs well.

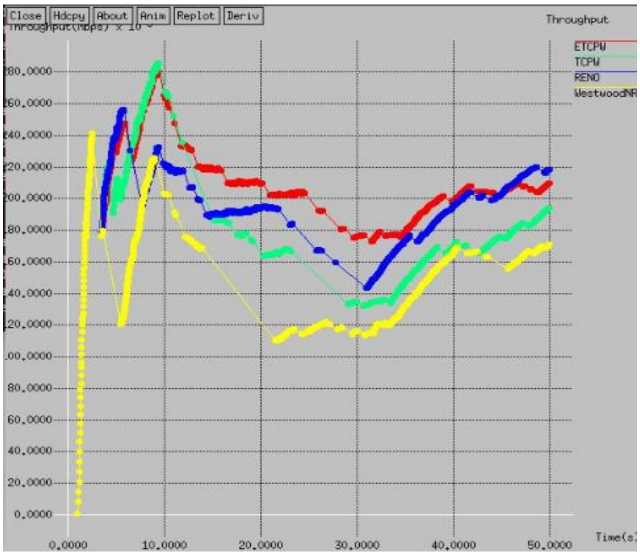

Fig.3. Throughput Comparison of Different Algorithms for error rate 0.13

Table-III: Average delay of different algorithms with variation of error rate

Error rate	Average Delay (Sec)			
	ETCPW	*TCPW*	*TCP Reno*	*TCP WestwoodNR*
0.03	0.2116	0.2356	0.2378	0.2268
0.13	0.1726	0.1597	0.2097	0.2936
0.23	0.1884	0.1367	0.1461	0.2411

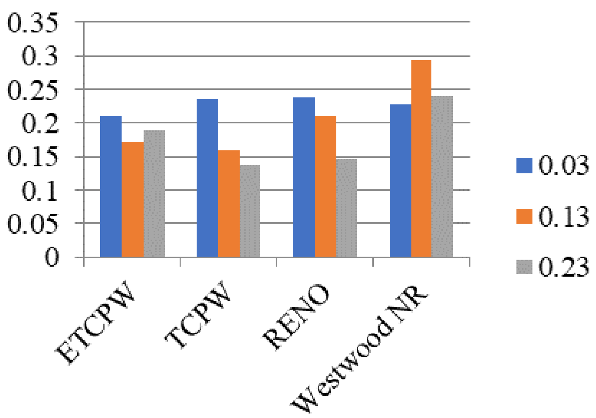

Fig.4. Average delay of different algorithms with variation of error rate.

TCP Westwood DCC has reduced delay as compare with other algorithms. Table-III and Fig.4 shows this comparison. This indicates in case of time out or link failure Westwood -DCC performs well and gives minimum delay. Enhanced Westwood-DCC has been compared with other TCP variants for PDR. Table IV shows the comparison of PDR.

Table -IV: PDR of different algorithms with variation of error rate

Error rate	PDR (%)			
	ETCPW	*TCPW*	*TCP Reno*	*TCP WestwoodNR*
0.03	94.9091	95.0631	95.3626	95.0198
0.13	84.6076	83.599	84.7351	84.6977
0.23	71.841	70.8808	71.3267	70.401

V. CONCLUSION

TCP Westwood DCC estimates the bandwidth and calculates ratio. As per the calculated ratio and network status it adjusts the cwnd. The proposed algorithm works for link failure scenario. The purpose of designing Westwood-DCC is to control congestion window during link failure in MANET. In Reno and other compared algorithms this adjustment has not been done and even these algorithms cannot handle link failure losses. TCP Westwood DCC has been implemented with Ubuntu and NS2.35. The performance is measured in terms of throughput, average delay and PDR. It has been analyzed that when we increase error rate TCP Westwood DCC performs well among other compared algorithms. Here TCP Westwood DCC is implemented for MANET, it can be further tested for wireless network also in the future.

REFERENCES

1. A. Ghaleb-Seddik, Y. Ghamri-Doudane, and S. M. Senouci, "Coupling Loss and Delay Differentiation to Eenhance TCP Performance within Wireless Multi-hop Ad-hoc Networks," J. Commun., vol. 7, no. 12, pp. 859–872, 2012.
2. M. Amin, K. Fard, S. Karamizadeh, and M. Aflaki, "Enhancing Congestion Control To Address Link Failure Loss Over Mobile Ad-Hoc Network," Int. J. Comput. Networks Commun., vol. 3, no. 5, pp. 177–192, 2011, doi: 10.5121/ijcnc.2011.3513.
3. M. A. K. Fard, S. Karamizadeh, and M. Aflaki, "Packet loss differentiation of TCP over mobile ad hoc network using queue usage estimation," 2011 IEEE 3rd Int. Conf. Commun. Softw. Networks, ICCSN 2011, pp. 81–85, 2011, doi: 10.1109/ICCSN.2011.6013667.
4. J. Ye, J. Wang, and H. Liu, "An adaptive loss differentiation algorithm based on queue management," Int. Conf. Commun. Technol. Proceedings, ICCT, no. 60673164, pp. 2–5, 2006, doi: 10.1109/ICCT.2006.341917.
5. C. Lim and J. Jang, "An Adaptive End-to-End Loss Differentiation Scheme for {TCP} over Wired/Wireless Networks," IJCSNS Int. J. Comput. Sci. Netw. Secur., vol. 7, no. 3, pp. 72–83, 2007.
6. S. Goswami, "END TO END LOSS DIFFERENTIATION SCHEME FOR IDENTIFICATION OF PACKET LOSSES IN TCP OVER WIRED & WIRELESS NETWORKS," no. 03, pp. 166–174, 2015.
7. M. Y. Park, S. H. Chung, and P. Sreekumari, "Estimating rate of queue usage to differentiate cause of packet loss in multi-hop networks wireless," Proc. - Int. Comput. Softw. Appl. Conf., vol. 1, pp. 500–508, 2009, doi: 10.1109/COMPSAC.2009.73.
8. S. Bregni and D. Caratti, "Enhanced loss differentiation algorithms for use in TCP sources over heterogeneous wireless networks," …, vol. 00, no. C, pp. 2–6, 2003.
9. V. P. Reshamwala and S. V Reshamwala, "TCP Westwood-NRAS Congestion Control Protocol with Dynamically Set Slow Start Threshold," vol. 2, no. 7, pp. 256–261, 2016.
10. M. Park, J. H. Choi, and H. Choo, "Enhanced Westwood as per Vegas-based Estimator with Slowstart Threshold for High-Speed Network s," pp. 92–99, 2008, doi: 10.1109/ICCSA.2008.30.
11. R. A. Hamamreh and M. J. Bawatna, "Protocol for Dynamic Avoiding End-to-End Congestion in MANETs," J. Wirel. Netw. Commun., vol. 4, no. 3, pp. 67–75, 2014, doi: 10.5923/j.jwnc.20140403.01.

12. A. Capone, L. Fratta, and F. Martignon, "Enhanced bandwidth estimation algorithms in the TCP congestion control scheme," IFIP Adv. Inf. Commun. Technol., vol. 107, pp. 469–480, 2003, doi: 10.1007/978-0-387-35620-4.
13. V. I. Kanani, "Performance Analyses of TCP Westwood," vol. 2, no. 2, pp. 1746–1750, 2014.
14. P. Kaushika and R. J. M, "A survey on effectiveness of TCP Westwood in mixed wired and wireless networks," no. March 2016, 2013.
15. S. Hagag, "Enhanced TCP Westwood Congestion Avoidance Mechanism (TCP WestwoodNew)," vol. 45, no. 5, pp. 21–29, 2012.
16. S. Thilagavathi and J. Vanitha, "A Survey on TCP Westwood," vol. 3, no. 6, pp. 233–237, 2014.
17. M. Al-hasanat, K. Seman, and K. Saadan, "Enhanced TCP Westwood Slow Start Phase," 2014, doi: 10.14738/tnc.25.601.
18. G. Yang, R. Wang, M. Y. Sanadidi, and M. Gerla, "TCPW with bulk repeat in next generation wireless networks," IEEE Int. Conf. Commun., vol. 1, no. June, pp. 674–678, 2003, doi: 10.1109/icc.2003.1204260.
19. S. Mascolo, C. Casetti, M. Gerla, S. Lee, and M. Sanadidi, "TCP Westwood: congestion control with faster recovery," Univ. California, Los Angeles, Tech. Rep. CSD TR, vol. 200017, pp. 1–14, 2000.
20. A. Barad, "Enhanced TCP WestwoodNR Congestion Avoidance mechanism (WestwoodNRBWP) in VANET," pp. 28–34, 2016.
21. K. S. Mohanad Hasanat Kamruzzaman Saman, "Enhanced TCP Westwood Congestion Control Mechanism over Wireless Networks," Int. Conf. Adv. Technol. Sci., no. October 2017, 2014.

AUTHORS PROFILE

Sejal Haveliwala received B.E. Degree in computer from Sardar Patel University, V, V Nagar, Gujarat, India, and pursuing Pd.D degree in Computer engineering from C. U. Shah University Surendranagar, Gujarat, India. She has been with Shree M.M Patel Institute of Sciences and Research, Dept. of M.Sc.IT, Gandhinagar ,Gujarat ,India since 2008 .Her research interests include mobile adhoc networking and communication especially for the performance evaluation of TCP in MANET, analysis, simulation.

Dhaval Vyas has been with C. U. Shah University, as Dean, Master of Philosophy and Academic officer. He has 14 years of experience as an Assistant Professor at MCA Department there. He Published 20 research papers in National/International Journals. He guided 7 M.Phil. and 3 Ph.D scholars. He has written 7 books in faculty of computer science and engineering.

Hemal Shah Completed Ph.D.in computer engineering from Ganpat University, Mehsana, Gujarat, India. He has done his research in field of Adhoc Network Routing issues which primarily focus Delay Tolerant Network, Routing Issues, Mobility Modeling. He has been with Ganpat university since last 18 years as Professor and head at ICT department.