MODIFIED CONGESTION CONTROL ALGORITHM IN TCP WESTWOOD FOR MOBILE AD-HOC NETWORK ENVIRONMENT

Anandakumar Haldorai

Associate Professor,Department of Computer Science and Engineering, Sri Eshwar College of Engineering, Coimbatore,India.

Abstract - TCP (Transmission Control Protocol) is a transport protocol that specifies how a network connection can be established and maintained by the end-to-end devices. Congestion is not managed correctly in the mobile ad-hoc environment. The congestion window cannot be controlled according to the type of failure and excessively reduces the congestion window size and diminishes the efficiency. TCP Westwood cannot recognize congestion or lack of connection failure and cannot handle the congestion window according to the transmission capacity. In this article, the Modified Congestion Control Algorithm in TCP Westwood (MCC-TCPW) is proposed that manages a congestion window and slow start thresholds based on the network available bandwidth. Connection failures can occur regularly in mobile ad-hoc networks and must be carefully managed. With the assistance of retransmission timeout, connection failure can be detected. Once Westwood has a timeout, it will prevent congestion. The proposed MCC-TCPW handles three congestion conditions 1) Prevention, 2) congestion, 3) No congestion, the heavy traffic window, and the slow start threshold is modified as per the network architecture. The window is automatically adjusted based on congestion. The network status determines the bandwidth and fairness ratio. The proposed Modified Congestion Control Algorithm in TCP Westwood (MCC-TCPW) is reviewed on NS 2, and the available TCP variants are compared. The proposed system optimizes the use of connections and the congestion management system. It delivers substantial recovery efficiency.
Keywords: TCP, Mobile ad-hoc network, congestion control, Westwood TCP, Transport protocol

1 INTRODUCTION

Mobile ad-hoc network (MANET) experiences packet loss issues due to congestion, connection failure, or wireless channel fault. TCP is a link-driven and trustworthy protocol [1]. It operates on recognition standards but cannot correctly distinguish packet loss. TCP should be configured so that a proper form of failure can be detected. TCP monitors algorithmic shortcomings [2]. When bandwidth is scarce, and internet traffic increases as space congestion occur. The TCP operates in two countries 1) sluggish beginning and 2) prevention of congestion [3].

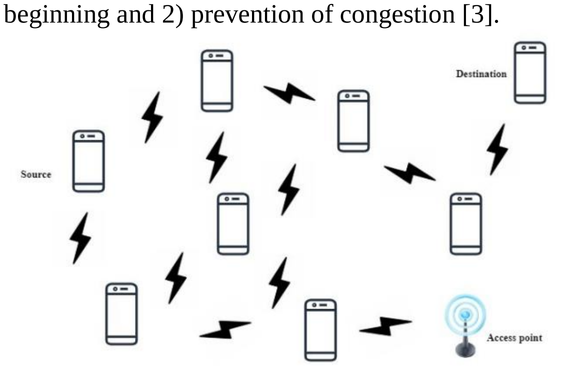

Figure 1 Mobile ad-hoc network architecture

Fig. 1 shows the architecture of the mobile ad-hoc network. Where each node can send, receive and transmit. The congestion window would be increased to the set threshold in the slow beginning and then moved to prevent TCP from turning the congestion window (cwnd_) to half [4]. It is noted that the error causes losses in wireless or MANET in the connection, and TCP decreases the loss as congestion. TCP operates algorithms such as slow start, fast transmission, fast retransmission, and multiplying increase and reduces to prevent congestion [5]. TCP cannot adequately manage congestion and cannot identify losses due to route or channel disruption. To work on this, TCP added several variations.

The second problem with TCP is the congestion management bandwidth calculation [6]. TCP must adequately estimate this to be cwnd_ and ssthresh_ during congestion or congestion control. TCP implemented several variants to manage congestion, such as TCP Tahoe, Reno, New Reno, Sack, and Westwood [7]. There is no bandwidth estimation in Tahoe. It detects three duplicate recognition of packet loss and transmits quickly [8].

Reno also lowers the congestion window without taking into account losses. It results in low performance [9]. Random and congestion failures cannot vary by Reno. Reno works well compared to single packet loss and not managing multiple packet losses, while New Reno will work on numerous packet losses [10]. New Reno doesn't wait for three duplicate acknowledgements until a missed packet is retransmitted, and Reno and New Reno will halve the limit and reduce the area to one [11]. New Reno also retransmits quickly and quickly to address the Reno crisis [12].

A recipient selectively accepts packets from the order in the sack instead of delivering all the packages [13]. This server detects and transmits incomplete packets instead of all unrecognized packets. In the case of a multiple packet failure, sack also provides better results [14]. Westwood uses battery-powered bandwidth and acts stably, thereby improving its efficiency. According to the bandwidth required, TCP Westwood doesn't handle half cwnd_ like TCP, but slow start threshold (ssthresh_). It's an algorithm dependent on rates [15].

The bandwidth is measured, and network services are efficiently used [16]. This paper proposes a framework for the complex controls of congested and non-congested network windows following available bandwidth and network positions [17]. The work is comparable to TCP versions such as Reno, Westwood, and Westwood NR.

The rest of the article as follows. Section 2 deals with the background and literature survey of the TCP Westwood. The proposed Modified Congestion Control Algorithm in TCP Westwood (MCC-TCPW) is designed and implemented in section 3. The software analysis and performance evaluation are illustrated in section 4. The conclusion and future scope are discussed in section 5.

2 BACKGROUND TO TCP WESTWOOD

Several researchers have experimented with developing the TCP varieties in the field of congestion prevention and avoidance processes. Both versions use various techniques for congestion management and vary when the failure to link occurs [18]. The work offered multiple approaches to TCPW enhancement and loss recovery.

2.1 Westwood NARAS

This system controls the congestion through the bandwidth assessment and prevents excessive city and ssthresh_ decreases [19]. If three Duplicate ACKs are received, the available network bandwidth is high, and the error is random. Bandwidth here as BW=BWE*RTT is predicted. In a TCP Westwood wireless connection, congestion management is performed in the slow start process, increasing cwnd_ at a reduced pace when the bandwidth is sufficient [20]. Westwood thus efficiently uses bandwidth and improves efficiency.

2.2 Slow Start Process: Enhanced TCP Westwood

A new slow start approach is specified to use a bandwidth estimate. The connection capacity is estimated to reflect the link utilization is expressed in Equation (1),

$$ELC = \frac{A_{seg_{size}}}{t_k - t_{k-1}} \quad (1)$$

ELC is denoted the estimated link capacity. $A_{seg_{size}}$ is denoted the average segment size. t_k is denoted the current RTT and t_{k-1} is denoted the previous RTT [21]. The algorithm employs an average sliding window approach to adjust ELC for each ACK is expressed in Equation (2)

$$ELC = (1-\propto)ELC_i + \propto ELC_{i-1} \quad (2)$$

α =0.9, ELC_i is denoted the current estimated link capacity and $ELC_{i-1}$is denoted the previously estimated link capacity [22]. Then slow start threshold ssthresh_ is expressed in Equation (3)

$$ssthresh = ELC * RTT_{min} \quad (3)$$

Where minimum round-trip time is RTT_{min}, this ssthresh_ is based on the connection capability measured [23]. The algorithm also monitors the sluggish start of cwnd_ values and the last RTT. In this condition, it sets the cwnd_ to the initial value. It modifies the TCPW's efficiency.

2.3 TCPW with Bulk Repeat

Three changes to the bulk replicate are 1) Massive retransmission: automatically retransmit unrecognized packet when failure is identified. 2) If a successive failure happens, a fixed timeout is enabled. 3) The window size is set in case of failure, the cwnd_ is retained to its initial value. In case of more failure, TCPW (BR) performs well. Does the bulk broadcast ask concerns such as how long do broadcasters relay in a window? Is the window for missing packets advanced? The TCPW technique is designed to resolve these issues.

It is quickly recovered if a sender has three times the recognition or retransmission period.

It sends unrecognized packets in a congestion window and missing packages when no congestion occurs. It fixes Retransmission Time Out (RTO) for a specified timeout. Here cwnd_ and ssthresh_ are configures to detect failure. When it is more significant than ssthresh_, the cwnd_ is not diminished. More data can also be transmitted if RTT happens due to the big cwnd_. Instead of reducing the cwnd_ size, it sets as ssthresh_. If the loss is found and not the congestive failure is more prominent than ssthresh_. It then works for better results [24]. By setting the congested window to 1 after a timeout, Westwood executes quick recovery. TCP Reno manages fast recovery in almost the same manner, which is a reasonable way to prevent the decrease from the congestion window to 1 in failures due to disturbance with the Wireless Connection and not overloading estimated bandwidth (BW_E) information is not used in its entirety.

2.4 TCP Westwood New

The enhanced congestion-prevention mechanism uses data receiving thresholds to forecast network status. The actual bandwidth is estimated, and the bandwidth proportion is defined. Suppose the ratio is below 1, network load increases. If the ratio is above 1, network load decreases. Here cwnd_ is set by the topology of the system. It helps to reduce congestion efficiently. It does the same as there is a timeout or three duplicate acknowledgement conditions. The algorithm for Retransmission Time Out (RTO) calculations is changed, too. Expiry of the RTO can also imply congestion problems [25]. TCP begins slowly during congestion. If RTO happens with this incorrect slow start capability and efficiency can also be decreased [26]. RTO shall be updated when a failure occurs, by New RTO is expressed in Equation (4)

$$RTO_{new} = \frac{RTT_{new}}{RTT_{old}} RTO_{old} \quad (4)$$

The new and old RTT are denoted as $RTT_{new}\ and\ RTT_{old.}$ The old and new RTO are expressed as $RTO_{old}\ and\ RTO_{new}$. It allows retrieving the loss of the connection. TCP Westwood works well for better performance and lower latency [27].

2.5 Enhanced Westwood

It was proposed to TCP to predict bandwidth when timeouts occur and to drop packages. The adjusted timeout is specified in this case to achieve better results than TCP Westwood (TCPW) [28]. The calculated bandwidth measurements are first verified with the last RTT before ssthresh_ and cwnd_ changes. The method computes the cwnd_ for three conditions 1) Congestion, (2) no congestion, and (3) reduction of congestion. The central concept is to verify the present position of the relation by two indicators until the timeout ends [29]. 1) Bandwidth approximate 2) the RTT_{last} which is the last RTT. Suppose the loss of a packet is $cwnd\_ < BW_E/2$, where BW_E is denoted the estimated bandwidth, so no congestion is indicated [30]. If the $RTT_{last} < RTT_E$ means a strong link. It should increase the cwnd_ by one section rather than decreasing it. The bit rate was boosted after TCPW. Where RTT_{last} is denoted the las RTT.

Based on the research background, the available TCP variants either try to control the congestion window or increasing the throughput. But these two parameters are inversely related. So Modified Congestion Control Algorithm in TCP Westwood (MCC-TCPW) is proposed to control the cwnd_ rate and the congestion decrement level.

3 PROPOSED MODIFIED CONGESTION CONTROL ALGORITHM IN TCP WESTWOOD (MCC-TCPW)

The algorithms listed above have performed on congestion management and bandwidth approximation. The congestion window has been updated based on network state: Westwood has used the receivers of bulk retransmissions. NARAS has experimented by recognizing the network state and conducting bandwidth calculation to prevent congestion. Enhanced Westwood works in a sluggish stage. It evaluates the slow-start thresholds according to the expected bandwidth. It functions well in a slow beginning process. Westwood observes failure during mass transmission, and ssthresh_ and cwnd_ are set. It also boosts performance and quickly recovers. Westwood New also carries out congestion avoidance according to the topology of the system.

RTO has been used to quantify the loss and to improve efficiency. Westwood has worked similar to new Westwood but has performed better for Westwood NR and has a network capability with decent performance for cwnd_ and ssthresh_. The TCP petra is improved in Westwood, which calculates bandwidth according to the existing state of the connections. The last RTT technique requires ssthresh_ and cwnd_ as well.

Many of the above methods have tried to monitor the state of the network or measure the bandwidth. The methodology proposed contains both the Westwood and the RTT_{last} variables are used to observe the system's status and set ssthresh_ and cwnd_. The algorithm worked according to the connection status for all these three conditions, such as prevention, congestion, and no congestion.

3.1 Modified Congestion Control Algorithm in TCP Westwood (MCC-TCPW)

The algorithm operates with the RTT_{last} and Bandwidth Product (BWP) has been used to calculate the bandwidth estimate. In comparison with other TCP versions, the effect will be on an increasing number of active transactions. The suggested approach is used to deal with TCP Westwood performance losses and development.

Cwnd_ is one of the essential variables for transmitting the quantity of data throughout the cycle. It also determines whether or not the connection is congested and how to prevent it. To determine the bandwidth ratio, present and former bandwidth estimates are the critical parameters. In the calculation of ssthresh_, the calculated bandwidth is utilized, which usually determines congestion level. In determining whether the network is in good condition or not, the last RTT and estimated RTT are used.

The fundamental concept is to identify connection status by 1) expected bandwidth, 2) last RTT, and 3) bandwidth product (BWP) after the deadline has expired. TCP Petra has not used ssthresh_ to prevent congestion. Both situations $cwnd\_ < BW_E/2$ and $BWP > 1$ are checked using the proposed method, under which the connection is well maintained and no congestion is present. It also scans the average value of capacity and the last RTT to verify the connection state to prevent congestion.

It checks whether $BWP < 1$ can send additional packets, so it needs to keep cwnd_ or set cwnd_. It does so by checking BWP. The methodology suggested works to estimate bandwidth during the loss of connection and increase performance compared to other versions of TCPs like Reno, Westwood, and Westwood NR.

3.2 The Significance of the Variables using the Input

The following variables have been considered in the paper,

$seg\_size\_$ - package size in bits.

BWP – is a delay and preceding bandwidth proportion. This proportion allows the link capacity to be tested. In case $BWP < 1$, the channel cannot send additional packets of data. $BWP > 1$ ensures that the channel will send packets. In addition to the RTT_{last}, an approximate bandwidth is required to verify network status. $RTT_{last} < RTT_E$ means the link is in acceptable condition, and it should increase the cwnd_ rather than restore it. It identifies three congestion conditions, BWP, $RTT_{last}$and RTT_E. RTT_E is determined by Equation (4)

$$RTT_E = RTT_t * tcp\_tick\_ \qquad (4)$$

Here RTT_t is the number of signals that were sent before $tcp\_tick\_$ command. The estimated RTT is denoted as RTT_E.

Fig. 2 shows the packet header format of the transmission control protocol. It contains the source address, a destination address, source port number, destination port number, flag, checksum, etc.

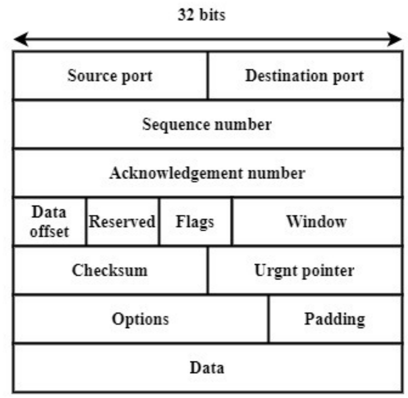

Figure 2 Packet header format of transmission control protocol

3.3 Steps in MCC-TCPW algorithm

1. Set variables for inputs such as last BW_E sample, actual BW, RTT_{min}, $seg\_size\_$, BW_E.
2. Timeout check for broadcast. (It's due to the failure of the link).
3. Measure the ssthresh_ then estimate bandwidth product (BWP) according to the usable bandwidth using Equations (5) and (6)

$$ssthresh\_ = \frac{BW_E * RTT_{min}}{seg\_size\_} \qquad (5)$$

$$BWP = \frac{BW_{current}}{BW_{previous}} \qquad (6)$$

The segment size is denoted as $seg\_size\_$. RTT_{min} is denoted as the minimum RTT. The current and previous bandwidth is represented as $BW_{current}$ and $BW_{previous}$, respectively.

4. Check the state of the network and use Equations (7), (8), and (9) to the congestion management framework. RTT_E is calculated by checking the status of BWP and last RTT.

i) In congestion-free state using BW_E & BWP. Set cwnd_

$$cwnd\_ = BW_E/2 \qquad (7)$$

ii) In Congestion avoidance state, set cwnd_ as ssthresh_

$$cwnd\_ = ssthresh\_ \qquad (8)$$

iii) In Congestion state if BWP < 1, set cwnd_

$$cwnd\_ = cwnd\_ + 1/cwnd\_ \qquad (9)$$

Where $cwnd\_$ denotes the congestion window size, $ssthresh\_$ denotes the slow start threshold value, BW_E is denoted the estimated bandwidth.

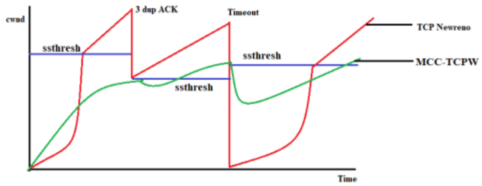

Figure 3 cwnd_ compares the proposed Modified Congestion Control Algorithm in TCP Westwood (MCC-TCPW) and existing TCP Newreno.

Fig. 3 shows the cwnd_ comparison of the proposed Modified Congestion Control Algorithm in TCP Westwood (MCC-TCPW) and existing TCP Newreno. As the congestion occurs, TCP Newreno reduces the cwnd_ without checking the actual congestion level of the network. But the proposed Modified Congestion Control Algorithm in TCP Westwood (MCC-TCPW) adjusts its cwnd_ based on the network congestion level by the value of BWP and last RTT.

The proposed Modified Congestion Control Algorithm in TCP Westwood (MCC-TCPW) is designed with the help of BWP and the last RTT to change the cwnd_ concerning the network congestion level. When congestion occurs, it performs well than the existing TCP Newreno protocol.

4 SIMULATION ANALYSIS AND PERFORMANCE EVALUATION

The algorithm's efficiency is enhanced in terms of performance and time due to three situations: congestion, congestion avoidance, and no congestion.

4.1 Evaluation Parameters

The assessment conditions are: (1) in the form of send number of data packets, how many are obtained per recipient, it is known as network performance. (2) delay, the time is taken for communication or transmission of a packet, is calculated in seconds. (3) The packet delivery ratio is the proportion of all data messages collected from all recipients to the number of senders' data packets. It is determined by Equation (10)

$$PDR = \frac{number\ of\ data\ packet\ received}{number\ of\ data\ packet\ sent} \qquad (10)$$

4.2 Simulation Setup

The specifications of the set simulation are seen in Table 1. Experiments with the NS2.34 simulator were performed in fedora. The root node here is five, and the target node is 18. The origin node transfers data packets across intermediate nodes to the target. The mistake model includes both the average packet process and policy. Here is using a uniform error model. 12 Mbps is allocated for bandwidth. The latency in spreading is 18ms. The traffic flow would take 600 seconds to simulate.

Table 1 Simulation parameters

Simulation parameters	Value
Simulation tool	NS 2
Mobility model	Random waypoint
Bandwidth	12 Mbps
Simulation area	250 m x 250 m
Mac type	IEEE 802.11
Antenna model	Omni antenna
Queue type	Drop tail
Error model	Uniform error model
Routing protocol	DSR
Simulation period	600 sec

4.3 Performance Analysis

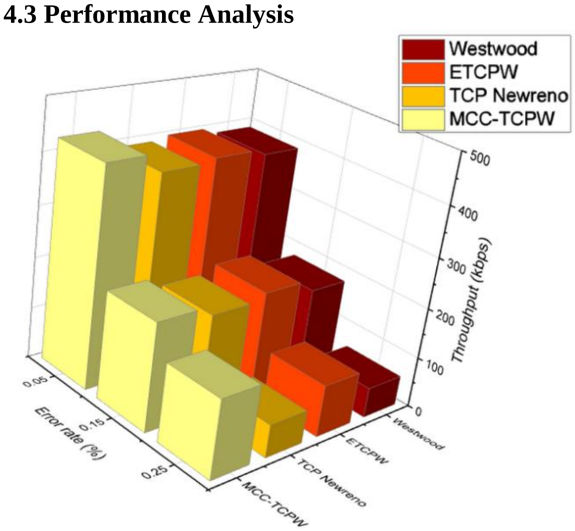

Figure 4 Throughput analysis of the proposed Modified Congestion Control Algorithm in TCP Westwood (MCC-TCPW)

Fig. 4 shows the throughput analysis of the proposed Modified Congestion Control Algorithm in TCP Westwood (MCC-TCPW). The TCP algorithms only varied, and the rest of the simulation parameters are maintained constant for the analysis. The throughput of the Westwood, ETCPW, TCP New Reno, and the proposed Modified Congestion Control Algorithm in TCP Westwood (MCC-TCPW) are analyzed. The results are plotted in the above Fig.s. The results show that the proposed Modified Congestion Control Algorithm in TCP Westwood (MCC-TCPW) has the highest throughput in all the error rates.

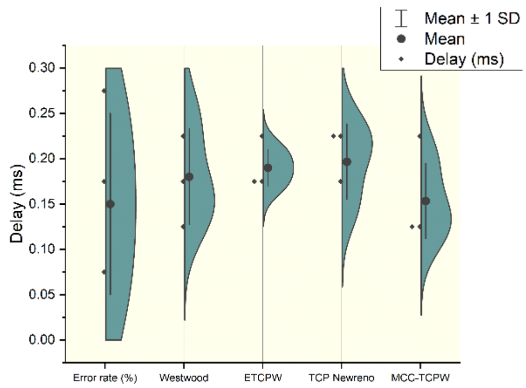

Figure 5 Delay analysis

Fig. 5 shows the delay analysis of the proposed Modified Congestion Control Algorithm in TCP Westwood (MCC-TCPW). The delay of the Westwood, ETCPW, TCP Newreno, and the proposed Modified Congestion Control Algorithm in TCP Westwood (MCC-TCPW) are analyzed. The results are plotted in the above Fig.s. The results show that the proposed Modified Congestion Control Algorithm in TCP Westwood (MCC-TCPW) has the lowest delay in all the error rates.

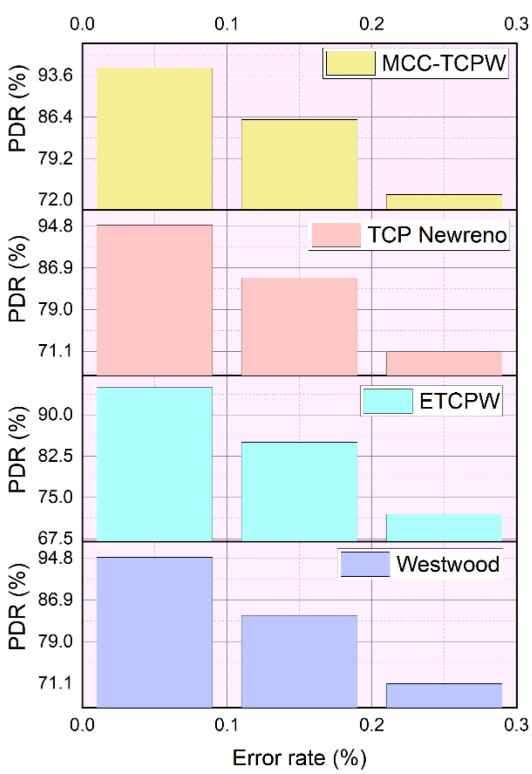

Figure 6 Packet Delivery Ratio analysis

Fig. 6 shows the packet delivery ratio analysis of the proposed Modified Congestion Control Algorithm in TCP Westwood (MCC-TCPW). The PDR of the Westwood, ETCPW, TCP Newreno, and the proposed Modified Congestion Control Algorithm in TCP Westwood (MCC-TCPW) are analyzed, and the results are plotted in the above Fig.s. The results show that the proposed Modified Congestion Control Algorithm in TCP Westwood (MCC-TCPW) has the highest PDR in all the error rates.

The proposed Modified Congestion Control Algorithm in TCP Westwood (MCC-TCPW) is designed and implemented. The simulation parameters such as throughput, delay, and packet delivery ratio are calculated. The proposed Modified Congestion Control Algorithm in TCP Westwood (MCC-TCPW) has the highest performance in all the scenarios.

5 CONCLUSION AND FINDINGS

Modified Congestion Control Algorithm in TCP Westwood (MCC-TCPW) measures and determines the frequency ratio of bandwidth. It varies the cwnd_ according to the estimated ratio and network position; the suggested algorithm functions for the connection loss scenario. The Modified Congestion Control Algorithm in TCP Westwood (MCC-TCPW) is designed to monitor the congestion window in the event of a MANET connection breakdown. This improvement has not been made, and even these techniques cannot manage losses of relation failure in Reno and other comparative algorithms. MCC-TCPW with fedora and NS2.34 have been adopted. The performance is calculated in terms of efficiency, average time, and PDR. the MCC-TCPW performs well compared to other protocols when enhancing the error rate. MCC-TCPW for MANET is proposed and can also be subsequently checked for cellular networks.

References

[1] Nigar, N., & Azim, M. A., “Fairness Comparison of TCP Variants over Proactive and Reactive Routing Protocol in MANET”, International Journal of Electrical and Computer Engineering, Vol. 8, no. 4, pp. 2199-2206, 2018.

[2] Wheeb, A. H., Morad, A. H., & Tameemi, M. I. A., “Performance evaluation of transport protocols for mobile ad hoc networks”, Journal of Engineering and Applied Sciences, Vol. 13, no. 13, pp. 5181-5185, 2018.

[3] Al-Akaidi, M., “A review on transport layer protocol performance for delivering video on an ad-hoc network”, IOP Conference Series: Materials Science and Engineering, Vol. 237, no. 1, 2017.

[4] Shenoy, S. U., Sharmila Kumary, M., & Shenoy, U. K. K., "Modified Transmission Control Protocol Reno to Improve the Performance in Mobile Ad Hoc Networks", Journal of Computational and Theoretical Nanoscience, Vol. 16, no. 4, pp. 1685-1689, 2019.

[5] Yadav, A. K., & Kush, A., "TCP-and UDP-based performance evaluation of AODV and DSR routing protocol on varying speed and pause time in mobile Ad Hoc networks", Next-Generation Networks, 2017.

[6] Kumar, S., Dutta, K., & Garg, A, "FJADA: Friendship Based JellyFish Attack Detection Algorithm for Mobile Ad Hoc Networks", Wireless Personal Communications, Vol. 101, no. 4, pp. 1901-1927, 2018.

[7] Ul Rashid, F., "Congestion Control Analysis Over Wireless Ad Hoc Networks", IUP Journal of Computer Sciences, Vol. 11, no. 3, 2017.

[8] Das, N., Bisoy, S. K., & Tanty, S., "Performance Analysis of TCP Variants Using Routing Protocols of MANET in Grid Topology", Cognitive Informatics and Soft Computing, 2018.

[9] Govindarajan, J., Vibhurani, N., & Kousalya, G., "Enhanced TCP NCE: A modified non-congestion events detection, differentiation, and reaction to improve the end-to-end performance over manet", Progress in Intelligent Computing Techniques: Theory, Practice, and Applications, 2017.

[10] Jude, M. J. A., Diniesh, V. C., Shivaranjani, M., & Shanju, R., "A feedback-aware reliable transport protocol with improved window increment mechanism for the inter-vehicular wireless network", WirelessPersonal Communications, Vol. 98, no. 1, pp. 1119-1134, 2018.

[11] Volkov, A. S., Bakhtin, A. A., Ugrovatov, A. V., Volkova, E. A., & Shalyagin, D. V, "Applying TCP-based protocols for mobile ad-hoc networks with PN signals in NS-3", 2018 Systems of Signal Synchronization, Generating and Processing in Telecommunications, 2018.

[12] Vivekananda, G. N., & Chenna Reddy, P., "Performance evaluation of TCP, UDP, and SCTP in MANETS", ", ARPN Journal of Engineering and Applied Sciences, Vol. 13, no. 9, pp. 3087-3092, 2018.

[13] Zahra, S., & Nasir, M. U., "Fine Tuning of TCP/IP for Ad Hoc Networks", International Journal of Computer Science and Information Security, Vol. 15, no. 3, pp. 398-403, 2017.

[14] Singh, A., Singh, R., & Gupta, S., "Evaluating the Performance of TCP over Routing Protocols in MANETs Using NS2", International Journal of Scientific Research in Network Security and Communication, Vol. 6, no. 4, pp. 1-4, 2018.

[15] Jiang, H., Luo, Y., Zhang, Q., Yin, M., & Wu, C., "TCP-Gvegas with prediction and adaptation in multi-hop ad hoc networks", Wireless Networks, Vol. 23, no. 5, pp. 1535-1548, 2017.

[16] Sharma, V. K., & Kumar, M., "Adaptive congestion control scheme in mobile ad-hoc networks", Peer-to-Peer Networking and Applications, Vol. 10, no. 3, pp. 633-657, 2017.

[17] Matsuo, K., Sakamoto, S., Oda, T., Barolli, A., Ikeda, M., & Barolli, L., "Performance analysis of WMNs by WMN-GA simulation system for two WMN architectures and different TCP congestion-avoidance algorithms and client distributions", International Journal of Communication Networks and Distributed Systems, Vol. 20, no. 3, pp. 335-351, 2018.

[18] Rodríguez Herlein, D. R., Talay, C. A., González, C. N., Trinidad, F. A., Almada, L., & Marrone, L. A., "Burst Error Analysis Introduced in Multiple Traffic of Protocols TCP Reno, Cubic, Westwood and Vegas on a Model of Hybrid Topology", Journal of Computer Science & Technology, Vol. 19,no.1, 2019.

[19] Dzivhani, M., & Ouahada, K., "Performance evaluation of TCP congestion control algorithms for wired networks using NS-3 simulator", 2019 IEEE AFRICON, 2019.

[20] Kotian, D., Shetty, S., Begum, S., & Gadagkar, A. V, "Analysis of Various TCP Variants over MANET Routing Protocols", International Research Journal of Engineering and Technology, Vol.6, no. 2, 2019.

[21] Ameur, C. B., Mory, E., Cousin, B., & Dedu, E, "Performance evaluation of TcpHas: TCP for HTTP adaptive streaming", Multimedia Systems, Vol. 24, no. 5, pp. 491-508, 2018.

[22] Malekzadeh, M., & Ghani, A. A, "A Model for Behavioral Tendency of TCP Congestion Control Variants in LTE Cellular and 802.11 ac Networks", Journal of Telecommunication, Electronic and Computer Engineering, Vol. 10, no. 4, pp. 113-125, 2018.

[23] Zhang, J., Wen, J., & Han, Y., "TCP-ACC: performance and analysis of an active congestion control algorithm for heterogeneous networks", Frontiers of Computer Science, Vol. 11, no. 6, pp. 1061-1074, 2017.

[24] Babu, A. R., & Kumar, P. S., "Congestion Control using Fuzzy-based Adaptive TCP Window Adjustment", Indian Journal of Science and Technology, Vol. 12, no. 12, 2019.

[25] Vaseer, G., Ghai, G., & Ghai, D., "Novel intrusion detection and prevention for mobile ad hoc

networks: A single-and multi-attack case study", IEEE Consumer Electronics Magazine, Vol. 8, no. 3, pp. 35-39, 2019.

[26] Polshchykov, K. O., Lazarev, S. A., & Zdorovtsov, A. D., "Neuro-fuzzy control of data sending in a mobile ad hoc network", Journal of Fundamental and Applied Sciences, Vol. 9, no. 2S, pp. 1494-1501, 2017.

[27] Pokhrel, S. R., & Choi, J., "Improving TCP performance over WiFi for the internet of vehicles: A federated learning approach", IEEE Transactions on Vehicular Technology, Vol. 69, no. 6, pp. 6798-6802, 2020.

[28] Majeed, J. H., Habeeb, N. A., & Al-Azzawi, W. K., "Performance investigations of internet protocol versions for Ad-hoc mobile networks based on qualnet simulator", Indonesian Journal of Electrical Engineering and Computer Science, Vol. 21, no. 1, pp. 497-504, 2021.

[29] Cen, N., Jagannath, J., Moretti, S., Guan, Z., & Melodia, T., "LANET: Visible-light ad hoc networks", Ad Hoc Networks, Vol. 84, pp. 107-123, 2019.

[30] Patriciello, N., Núñez-Martínez, J., Baranda, J., Casoni, M., & Mangues-Bafalluy, J., "TCP performance evaluation over backpressure-based routing strategies for wireless mesh backhaul in LTE networks", Ad Hoc Networks, Vol. 60, pp. 40-51, 2017.