Journal of Theoretical and Applied Information Technology
15th November 2019. Vol.97. No 21

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SDCM: SECURE DYNAMIC END-TO-END CONGESTION AVOIDANCE PROTOCOL FOR MANETS

RUSHDI A. HAMAMREH[1]

Department of Computer Engineering, Faculty of Engineering, Al-Quds University, Jerusalem, Palestine

rhamamreh@staff.alquds.edu

ABSTRACT

Ad-hoc Mobile Networks (MANETs) are vulnerable to various attacks. In addition, congestion can occur due to limitation in resources and lead to high packet loss, long delay and waste of resource utilization time. The major objective of congestion control is to best utilize the available network resources by keeping the load below the capacity. The great demand for capacity, place particular emphasis on congestion management approaches.

Recently, researchers have developed many effective and well-studied algorithms for congestion control within Transmission Control Protocol (TCP) to improve its performance over MANET environment. TCP is designed to be reliable and ensure end to end delivery in wired network. However, each existing TCP variant over MANET has its weaknesses and strengths when changing MANET factors like: node mobility, traffic loads, network size and wireless channel conditions. In this paper a new approach to decrease packet loss using secure and dynamic path congestion estimation. The simulation results show an improvement in TCP performance and security over MANET in different scenarios.

Keywords:- *TCP-VEGAS; TCP-WESTWOOD; TCP-WELCOME; TCP-DCM; MANET; Congestion; Link Failure; Signal Loss, RTO.*

1. INTRODUCTION

The transmission control protocol (TCP) is the most predominant transport layer protocol in the Internet today. It is a reliable, end to end, connection oriented transport layer protocol that is split into segments. The major functions of TCP include congestion control, flow control, in order delivery of packets and reliable transportation of packets. Congestion control handles the overflow traffic in the network which leads to degradation in the performance of the network. TCP manages the number of packets sent to the network by increasing and decreasing the congestion window (*CWND*). The TCP sender starts the session with a congestion window value of one MSS.

The major function of Congestion Window (*CWND*) is to limit how much data allowed having in transit at a given time. The congestion window is congestion control‘s counterpart to flow control‘s advertised window that is received from the destination. TCP is modified such that the maximum number of bytes of unacknowledged data allowed is now the minimum of the congestion window and the advertised window [1] [1].

$$MaxWindow = \min(CWND, AdvertisedWindow)$$

Congestion Window (*CWND*) has to be calculated by sending side of TCP. Once the ACK is received within the retransmission timeout (RTO) period, the congestion window is doubled and this is called slow start.

Once it reaches the slow start threshold it grows linearly by adding one Maximum Segment Size (MSS)[2]to the congestion window every ACK received. This continues until packet loss detected which start congestion avoidance mechanism that reduces the slow start threshold to half the current CWND and reduce the congestion window size to one MSS. Since TCP is widely used, several mechanisms are developed to improve TCP's performance over Mobile Ad Hoc Networks[3].

In mobile ad hoc networks (MANET) communication is happen via wireless means and it can be heterogeneous wireless. There is no centralized controlling node since there is no preexisting Infrastructure, every node have to play the roles of both hosts and routers.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

MANET is the dynamic Topology which leads to frequent routing updates. Resources shared in MANET are mostly the bandwidth of the links and the queues on the routers or switches[4].

These special characteristics make some critical challenges to TCP since it was not originally designed to work in such environments, where the level of noise is not negligible due to the physical medium.

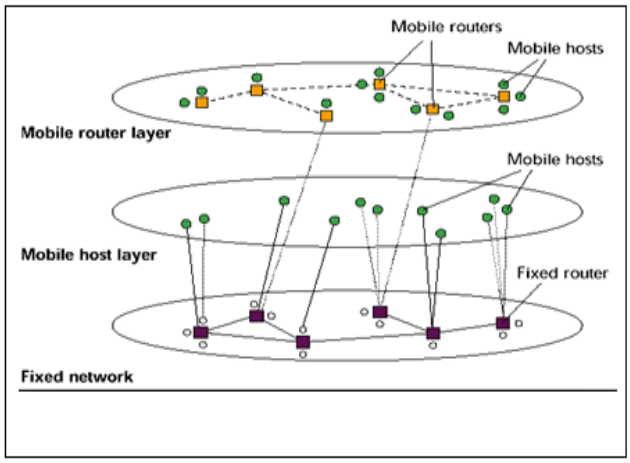


Figure 1: Mobile Ad Hoc Network (MANET)

Another problem is mobility that further degrades the performance of TCP by reduces its transmission rate whenever it detects a lost packet because TCP handles every packet loss as congestion, while in fact it's due to link failure. This lead to increase Retransmission Time Out (RTO) exponentially and remain high value even when find new route.

This paper is organized as follows. In Section II, related work of TCP-New Reno, TCP-VEGAS, TCP-Westwood and TCP-WELCOME with its algorithms, disadvantages. Section III presents the problems with existing Solution. Section IV presenting the proposal for solution. Simulation tools and validation Scenario with experimental results presented in section V. Section VI discuss the conclusion.

2. RELATED WORK

2.1. Existing TCP Variants

There are several variants of TCP implemented to solve the problems over MANET. In this paper I will focus on four main variants: TCP New Reno, TCP VEGAS, TCP WESTWOOD, and TCP WELCOME.

A. TCP NEW RENO

TCP New Reno is an effective modification of the original congestion avoidance algorithm in TCP RENO. The modification is an improvement of the Fast Recovery phase. CWND is modified as the following,

$$CWND_{new} = \begin{cases} CWND_{old} + 1 & CWND_{old} < ssthresh_{old} \\ CWND_{old} + \dfrac{1}{CWND_{old}} & CWND_{old} > ssthresh_{old} \end{cases} \quad (2)$$

When packet loss is detected, CWND and ssthresh are modified as,**Error! Reference source not found.**.

$$ssthresh_{new} = \frac{CWND_{old}}{2} \quad (3)$$

$$CWND_{new} = \begin{cases} ssthresh_{old} & \text{if three DUPACK received} \\ 1 & \text{if coarse timeout expires} \end{cases} \quad (4)$$

***Disadvantages*,**

TCP New Reno takes one RTT to detect each packet loss. In order to decide which segment lost then Ack of pervious transmitted segment must be received.

B. TCP-VEGAS

Vegas is an enhancement of RENO. It depends on proactive measure to encounter congestion in much more efficient than reactive ones. It overcomes the problem of requiring enough duplicate ACKs to detect a packet loss, and it suggests a modified slow star

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

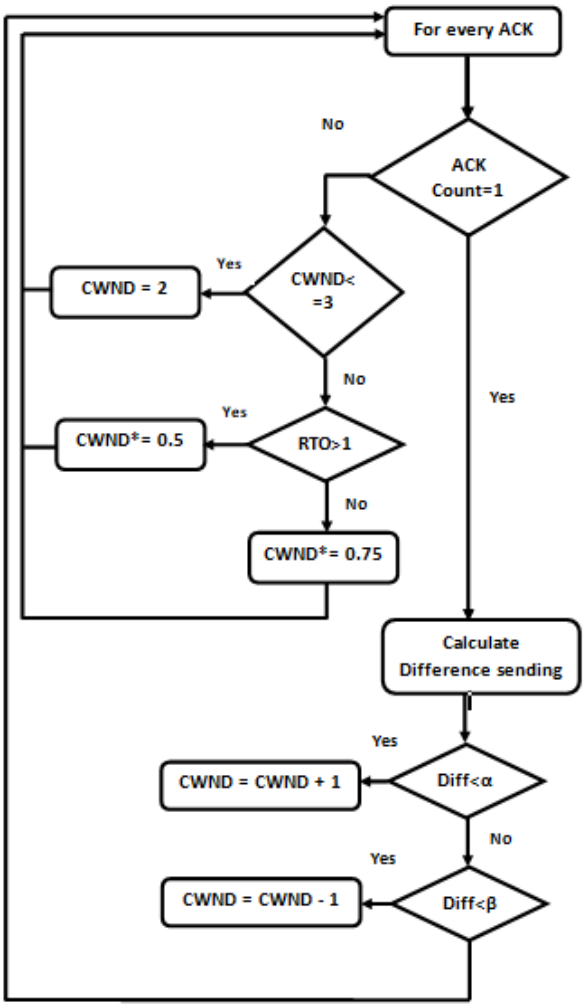


Figure 2: The flowchart for congestion control in TCP-VEGAS

Vegas is an enhancement of RENO. It depends on proactive measure to encounter congestion in much more efficient than reactive ones. It overcomes the problem of requiring enough duplicate ACKs to detect a packet loss, and it suggests a modified slow start algorithm which prevents it from congesting the network[6].

The new retransmission mechanism in Vegas extends on the retransmission mechanism of RENO. It keeps track by calculates an estimate of the RTT. TCP Vegas is different from all the other implementation in its behavior during congestion avoidance. It determines congestion by a decrease in sending rate as compared to the expected rate as the figure (2).

When new Ack is received,

$$\text{Actual sending rate} = \text{CWND}_{\text{last}} / \text{RTT}_{\text{last CWND}} \quad \textbf{(5)}$$

$$\text{Expected} = \text{CWND}_{\text{current}} / \text{RTT}_{\min} \quad \textbf{(6)}$$

$$\text{Difference} = \text{Expected} - \text{Actual} \quad \textbf{(7)}$$

$$\text{cwnd} = \begin{cases} \text{cwnd} + 1 & \text{diff} < \alpha \\ \text{cwnd} & \alpha \le \text{diff} < \beta \\ \text{cwnd} - 1 & \text{diff} > \beta \end{cases} \quad \textbf{(8)}$$

TCP Vegas increases *cwnd* linearly for the next RTT, if *Diff* <α and decreases *cwnd* linearly, if *Diff* >β. Otherwise, Vegas leaves *cwnd* unchanged [7].

When n duplicate Acks are received,

$$\text{cwnd} = \begin{cases} \text{cwnd} \times \frac{3}{4} & (\text{cwnd} > 3 \text{ and RTO} < 1) \\ \text{cwnd} \times \frac{1}{2} & (\text{cwnd} > 3 \text{ and RTO} < 1) \\ 2 & \text{cwnd} <= 3 \end{cases} \quad \textbf{(9)}$$

Disadvantages,

TCP-VEGAS performance decrease in the case of wrong RTT estimation, since it based on the value of RTT. TCP VEGAS performance also decrease when buffers at routers decrease.

C. TCP-WESTWOOD

TCP-Westwood uses bandwidth estimation to achieve protocol performance in mixed wired and wireless networks as following,

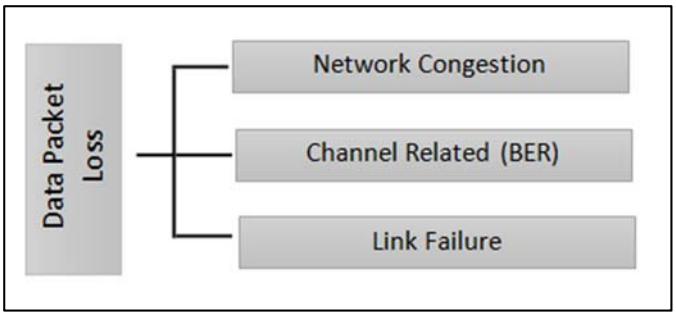
When new ACK reception, Congestion Window (CWND)is increased accordingly to the Reno algorithm; the end-to-end bandwidth estimate *BWE* is computed;[8].

$$\text{BWE} = \frac{\text{Ack\_Size}}{\text{Ack\_Interval}} \quad \textbf{(10)}$$

$$ssthresh = \max(2, \frac{\text{BWE} \times \text{RTT}_{\min}}{\text{SegSize}}) \quad \textbf{(11)}$$

Value of CWND is updated according to equation (4) as in TCP newReno. If packet loss is detected by three duplicated ACK ,then CWND = ssthresh.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195



Disadvantages,

Inability to differentiate between packet losses cause. This will degrade its performance since link failure is a major reason for packet loss. Performs poorly if it estimates incorrect Bandwidth.

D. TCP-WELCOME

TCP Wireless Environment, Link losses, and Congestion packet loss ModEls (WELCOME) [9] is a sender side based solution that improves the TCP performance by its ability to differentiate between causes of packet loss and then triggers the most appropriate packet loss recovery algorithm according to the identified loss cause.

Consist of two phases:

Loss Differentiation Algorithm:

Identify the reason of packet loss during the data transmission accurately. Packet loss causes in MANET:

a. Wireless Channel caused by several factors like: Signal Fading, Interference, Obstacles and environment effects

b. Link Failure caused by several factors like: Mobility, battery and obstacles

c. Congestion occurs at nodes buffer.

As in the following figurer:

Figure 3: Types of packet losses

TCP-WELCOME useround Trip Time (RTT) measurement to determine the cause of packet loss according to the following equation:

$$RTT(t) = 2\sum_{i=1}^{n}\left[q_{d,i}(t) + P_{d,i}(t) + p_{d,I}(t)\right] \quad (12)$$

Where: q is queuing time; P is the propagation time and p is the processing time

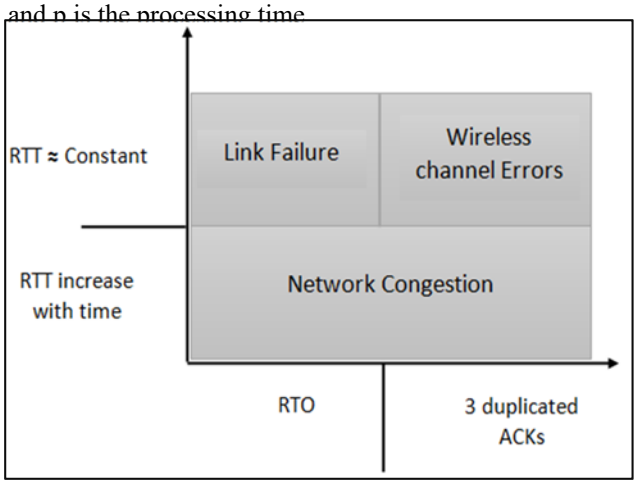


Figure 4: TCP WELCOME loss differentiation based on RTT, RTO and 3 duplicated Ack

As shown in figure 4, if value of RTT increase gradually then packet loss is due to congestion. This is because processing time at nodes buffers increase gradually. However, if value of RTT remains constant, then RTO or 3 duplicated Ack will decide. If RTO is triggered then the cause of packet loss is due to link failure. If 3 duplicated Ack received, then the cause of packet loss is wireless Chanel errors.

Loss Recovery Algorithm:

This algorithm triggers the appropriate recovery mechanism after identifying the cause of packet loss. There are three recovery cases in this TCP variant which are:

1- Network congestion related packet recovery algorithm. TCP-WELCOME use congestion

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

control algorithm of TCP New Reno for recovery of packet.

2- Link failure related packet loss recovery algorithm. TCP WELCOME adjust both values of RTO and CWND based on the ratio of RTT new and RTT old values as the following:

$$RTO_{new} = \frac{RTT_{new}}{RTT_{old}} \times RTO_{old} \quad (13)$$

$$CWND_{new} = \frac{RTT_{old}}{RTT_{new}} \times CWND_{old} \quad (14)$$

3. Wireless related packet loss recovery algorithm. TCP WELCOME does not make any changes, just retransmit the lost packets.

Disadvantages:

TCP WELCOME implements TCP New Reno for recovery when packet loss is identified as congestion loss.

2.2 MANETs Routing Protocols

Routing protocols are classified into three categories: Proactive, Reactive and Hybrid.

Table Driven (Proactive): In this type the routes to all destinations are determined at the start up and maintained by using a periodic route update process. The Advantage of this type is that routes always available. The Disadvantage of this type is the very high control overhead needed to maintain all routes.

Source Initiated (Reactive) [10]**:** In this type the route is determined only when it is required by the source, and it is maintained as long as it is needed. The Advantage of this type is the low control overhead needed since it is on demand. The Disadvantage of this type is the high initial delay needed to discover the route to destination.In this paper we will focus our work inAODVprotocol as oneof MANET routing reactive protocols.

AODV is a reactive routing protocol that uses next hop routing approach. Each node maintains a single path to a destination. AODV consists of two routing operations: Path discovery process and Path maintenance process. Every node maintains two separate counters: Sequence Numberand Broadcast ID.

Source node starts path discovery by broadcasting a route request (RREQ) packet to its neighbors, which includes source address; source sequence number; broadcast id; destination address; destination sequence number; and hop count. If the receiving node is an intermediate node with valid route to destination, it will send back a route reply (RREP) using the reverse path only if RREQ's ence number is smaller than that reco the intermediate node, or sequence numb equal with smaller hop count in the intermediate . Otherwise the intermediate nod oadcast the RREQ.

If the receiving node is the destination node itself, it will send back a RREP using the reverse path. A RREP contains the following information: source address, destination address, destination sequence number, hop count, and lifetime.

AODV stores only one route per destination with a certain life time. Once a route is established, it must be maintained as long as the route expiration time does not expire. This is done by exchanging "hello" packets periodically[11].

3. PROBLEM DEFINATION

In MANETs dropped packets due to various attacks or control flow of packets . The problem of TCP over Mobile Ad Hoc Network is applying congestion control algorithm to types of packet losses that are not lost due to congestion. When a packet is detected to be lost, either by timeout or by duplicated ACKs, TCP decrease the sending rate by adjusting its congestion window (CWND).

In wireless network bit error rates are very high things that cause packet losses due to physical channel. Another factor that exist in MANET is the dynamic topology due to node mobility, things that case packet loss from link failure till find new route. These kinds of packet loses are misjudged by TCP and treated as congestion, things that cause TCP performance degradation.

Many approaches use Round Trip Time (RTT) and Bandwidth (BW) estimation, but none of them work perfect in all scenarios without any problems. TCP WELCOME performs much better than other variants over MANET, because its ability to differentiate between different types of packet losses, However it apply the traditional congestion algorithm in TCP-New RENO. In this paper, a new

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

technique of congestion avoidance is proposed to replace the traditional congestion algorithm of TCP-NEWRENO.

4. PROPOSAL FOR SOLUTION

TCP-Welcome success to identify causes of packet losses as a key solution to the problems of TCP over MANET, However it has a weakness of applying conventional mechanism used by TCP-NewRENO of congestion control. To solve this problem, we now present our proposed Dynamic end-to-end Congestion detection protocol for MANET (**TCP-DCM**), a new cross layer solution that uses the results of route request process from routing protocol to early detect the end-to-end congestion, and dynamically select the path with minimum congestion from source to destination.

As mentioned in section II.2, during route request process, sender initiate a Route Request message to one hop nodes surrounding it. The process remains until this message receives the destination. After that destination select the shortest path and generate a Route Reply message to sender. Other valid paths from source to destination will be discarded.

Our modification has two phases: one on the destination node before generating Route Reply message and the other at sender side during the connection.

In first phase, destination will select three valid paths (if possible) which have the minimum cost, and send them to source node through Route Reply message.

In second phase, source will measure Round Trip time (RTT) under the control of TCP. After that source will select the minimum RTT path as main path from source to destination.

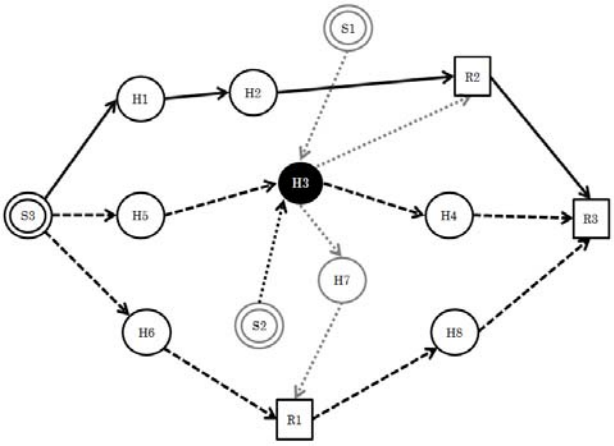


Figure 5: Ad hoc network with three TCP connections

During communication, if TCP detect gradual increase in RTT greater than congestion threshold level (CONGESTION_THREHOLD) as in equation (15), TCP will notify the network layer to check the validity of other paths available at source. If source found another path with lower RTT than current path, it will notify network layer to select that path for communication.

$$\text{CONGESTION\_THREHOLD} = RTT_{\min} * A \quad \textbf{(15)}$$

Value of factor A ranges between 1.7 to 1.95. Low values of A make unnecessary fluctuation between valid paths. High values of A will confuse the network between congestion and link failure.

If source node success to find another valid path with lower value of RTT, then it has to calculate the new values of CWND, RTO and ssthresh accurately. Selecting a high value of CWND will lead to congestion and cause more packet losses, on the other hand selecting a low value of CWND will decrease the throughput of the network. Estimated value of new ssthresh will be calculated depending on equations (10) and (11) that used by TCP-Westwood. Estimated value of new RTO is calculated by equation (13) that used by TCP-Welcome when selecting a new path. Our equations for estimating the value of CWND when selecting another path is as following,

$$CWND_{new} = \left(\frac{RTT_{new}}{RTT_{old}} \times ssthresh_{new} \right) \times B \quad \textbf{(16)}$$

Value of factor B ranges between 0.6 to 1. Low values of B will decrease the throughput. High

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

values of B will be better if the new estimated value of ssthresh is much higher than old one of the previous path.

Otherwise if neither existing paths valid nor have lower value of RTT than current one, TCP will update congestion window (CWND) as,

$$CWND_{i+1} = \begin{cases} \text{CWND}_i + \left(\frac{\text{RTT}_i}{\text{RTT}_{i+1}} \times 0.9\right) & \text{CWND}_i < \text{ssthresh} \\ \text{CWND}_i + \frac{\text{RTT}_{i+1}}{\text{CWND}_i \times RTT_i} & \text{CWND}_i > \text{ssthresh} \end{cases}$$

In order to avoid fast retransmission requests generated by the receiver node, sender will generate packet carries the sequence number of the segment at the head of the queue buffered at the congested hop and the reply packet have the sequence number of the last successful received segment at receiver node. TCP receiver will have the ability to understand the packets lost in transition and those buffered at the congested hops.

As an example shown in fig. 5, sender S3 try to find valid path to receiver R3. But in this network there are other two TCP connections: (S1:R1) and (s2:R2). S3 find three valid paths to R3 with minimum RTT: P1:(H1,H2,R2), P2:(H5,H3,H4), P3:(H6,R1,H8). Hop H3 is currently in congestion and used by S1 and S2. This will result with more end to end delay on path 2 due to congestion. S3 will use first path due to its lowest value of RTT. And store the other two paths with RTT value of each. Remember that we are dealing with dynamic topology; value of RTT will be used in future to determine the validity of the path to avoid waiting long time.

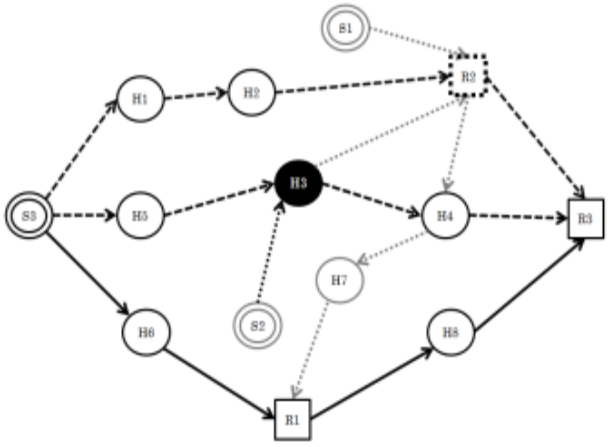


Figure 6: Ad hoc network with three TCP connections

As H3 hop become congested, (S1:R1) TCP connection will check alternative path to avoid using hop H3 as in Fig. 5. As a result R2 hop becomes congested. So (S3:R3) TCP connection will dynamically check the second and third alternative paths available at S3. If another path found to be valid and lower in RTT value, then S3 will select this path as primary path.

The average queue length can provide a direct measurement of the congestion status. The maximum value of the threshold (*Max*) and minimum value of the threshold (*Min*) are set to the queue length respectively, and the queue threshold represents the current state of the queue. Here, *Wq* stands for the queue weight, and the link utility would be very low if the three thresholds' values are set too small. On the other hand, congestion may occur before the node is notified if the thresholds are too large. Equation (18), (19), and (20) are used to set the thresholds' values [20].

$Min = 35\% \times$ Que_length **(18)**

$Max = Min \times 2$ **(19)**

$W_{qup} = W_{qpre}$ x H x S **(20)**

The average queue length is calculated by Equation (21) as followed.

Aver_queue = (1-*Wq*) x *Aver_queue* + *Ins_queue* x *Wq* **(21)**

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Equation (20) is used to dynamically set W_q. H represents hop counts, and S represents the number of packets sent per second. If the average queue (*Aver_queue*) length is smaller than the value of *Min* and the instant queue (*Ins_queue*) is smaller than half of the queue length, nodes are in the normal status [20].

The following is pseudo code for proposed algorithm:

```
Initialization
 //when Route-Reply message receive
For each path
        // notifying network layer Measure  RTT
End
Select path with minimum RTT
Estimate BW
Calculate new ssthresd and cwnd
Store  second and third pathswith each RTT  //if possible
```

```
Running
 // packet loss detected due to congestion
If currentRTT > RTTthreshold
 For each path
        // notifying network layer
  Measure RTT
    If measuredRTT < currentRTT
       Select path
      Estimate BW
      Calculate queue weight Wq
      Calculate Min, Max , Aver_queue
      Calculate new ssthresd and cwnd
      Generate packet with segments at congested nodes
 END
```

5. TOOLS, VALIDATION MODEL AND SIMULATION ENVIRONMENT

Simulation can carry out experiments without the actual hardware and provides a good compromise between complexity and accuracy[14][15]. In this section, I will present the performance metrics used in validation, simulation tools and experimental results.

5. 1. Performance metric

In order to evaluate algorithm effects on TCP performance, the following metrics must be considered [17][18][19]:

1. Packet Delivery Fraction: It is the ratio of the number of packets received successfully and the total number of packets sent.

$$\text{PDF} = \frac{\text{Packets Recieved Successfully}}{\text{Total Packets Sent}} \quad \textbf{(22)}$$

2. Throughput: It is the number of packets received successfully with respect to time.

$$\text{Throuputs(bits/s)} = \frac{\text{DeliveredPackets} * \text{PacketSize} * 8}{\text{TotalSimulationPeriod}} \quad \textbf{(23)}$$

3. Average end-to-end delay: The end-to-end-delay is averaged over all surviving data packets from the sources to the destinations.

4. The over head when the network is large.

$$\text{Overhead} = 1 - \frac{SentPackets}{SentPackets + ControlPackets} \quad \textbf{(24)}$$

5. 2. SIMULATION TOOLS

The proposed algorithm has been implemented and evaluated over NS 2 simulator which is discrete event simulator. NS 2 is written in C++, which is object oriented language. NS2 support simulation of different variant TCP and different routing protocols over wired and wireless networks.

Environment of implementation in this paper is done on size of 800*600, nodes concentration dense: (20,40,60,80)distributed randomly. We also generate 50 TCP connections between random senders and receivers. The average of three scenarios each has 30 random direction patterns of movement and random velocities between 0m/s to 3 m/s. packet size is 1460 byte.

Table 1: Simulation parameters.

MANET Parameter	Value
Value x	800
Value y	600
Simulation time	150s
Speed	(0-3) m/s
Routing protocol	AODV, DSDV
Mobility	Random
Maximum Connections	8
Number of nodes	20,40,60,80
Packet size	1460
Data Rate	1 Mbps
Traffic Type	Constant Bit Rate (CBR)
MAC Protocol	Mac/802_11

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

All results depend on output of resulting trace file generated from simulation.

The results of simulation are as the following:

Table 2: Average throughput (kbps) for node dense = 40.

Nodes dense = 40	
TCP Protocols	Average throughput (Kbps)
New Reno	417.8
VEGAS	374.2
WESTWOOD	409.2
WELCOME	431.1
DCM	439.4
SDCM	454.7

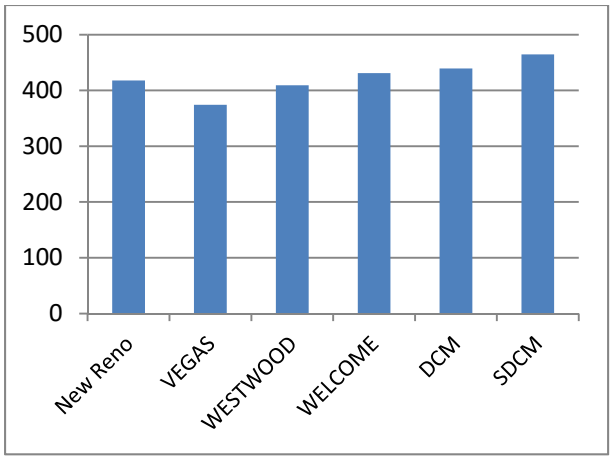


Figure 7: Average throughput (Kbps) for 40 nodes.

Table 3: Average throughput (kbps) for node dense = 20,40,60,80.

Nodes	Nodes dense		
	Welcome	SDCM	SDCM
100	440.2	464.1	484.4
80	449.7	467.3	485.3
60	452.6	462.2	478.2
40	431.1	439.4	454.7
20	425.3	426	426

The congestion may occur when the average queue length is between the values of *Min* and *Max*, and the discovery mechanism for a secure path is initiated. When the *Ins_queue* is larger than *Max*, the value of *Max* should be reset because of the wrong selected path. If the *Aver_queue* length is larger than *Max*, the nodes are in the congestion status and the congestion control is carried out. The flow chart of the algorithm is shown in Fig. 8.

TCP-DCM shows better results than TCP-WELCOME, due to its flexibility in treatment congestion problems over MANET. TCP-DCM shows increase in average throughput as node dense increase.

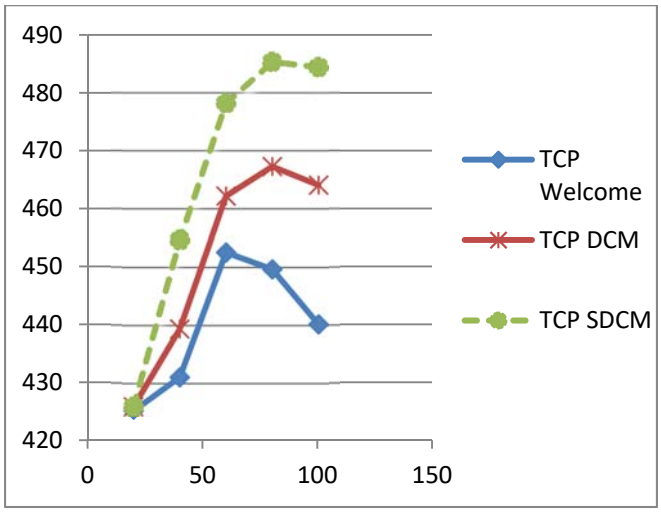


Figure 8: Average throughput (Kbps) for 20,40,60,80 nodes.

5. CONCLUSION

This paper presents current research on solving TCP problems over MANET by presenting most used TCP variants that preserve end to end semantic and there analysis to increase performance of TCP over MANET. We have placed special emphasis on TCP-WELCOME, because it is the most successful TCP variant over MANET, due to its ability to differentiate between types of packet losses in MANET. This article proposed a new mechanism to replace traditional congestion algorithm of TCP-NewReno used in TCP-WELCOME with dynamic minimum congestion path selection through cross layer analysis. With reference to data analysis and the experimental results, it shows that, TCP-DCM handles packet losses due to congestion in more efficient way than TCP-WELCOME does. Hence it improves overall throughput, increase TCP performance and avoidance attacks through selecting secure path the over MANET.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFERENCES

[1] Swastik Brahma and Mainak Chatterjee, *"Congestion control and fairness in wireless sensor networks", Pervasive Computing and Communications Workshops (PERCOM), 2010 8th IEEE International Conference*, March 29 2010-April 2 2010, pp413-418.

[2] Rushdi Hamamreh, Mohammed Bawatna , *"Protocol for Dynamic Avoiding End-to-End Congestion in MANETs". Journal of Wireless Networking and Communications.* 2014, 4(3): 67-75

[3] Ismail, D., Johor ; Jaafar, M., *"Mobile ad hoc network overview", Applied Electromagnetics 2007. APACE 2007. Asia-Pacific Conference,* Melaka, 4-6 Dec. 2007.

[4] Kumar, K.D. , Ramya, I. , Masillamani, M.R., *"Queue Management in Mobile Adhoc Networks (Manets)", Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on & Int'l Conference on Cyber, Physical and Social Computing (CPSCom),* Hangzhou, 18-20 Dec. 2010.

[5] Weinan Zhang, Li Lin, Li Du, *"The Study of Secure Congestion Control for TCP in Ad Hoc Networks".* *Journal of Information Security,* 2018, 9, Pp. 25-32.

[6] Basem Shihada, Qiong Zhang, Pin-Han Ho, *"A novel implementation of TCP Vegas for optical burst switched networks", Optical Switching and Networking,* Vol. 7, Issue 3, July 2010, Pp. 115–126.

[7] Li Huang ; Sch. of Comput. Sci. and Technol., Soochow Univ., Suzhou, China ; Jian-de Lu, *"Hops adaptive algorithm for TCP vegas in mobile ad hoc network", Mechatronic Science, Electric Engineering and Computer (MEC), 2011 International Conference, Jilin,* 19-22 Aug. 2011.

[8] Grieco, L.A. , Dipt. di Elettrotecnica ed Elettronica, Politecnico di Bari, Italy ; Mascolo, S., *"End-to-end bandwidth estimation algorithms for Westwood TCP congestion control", IEEE Information Technology Interfaces (ITI) 2003. Proceedings of the 25th International Conference*, 16-19 June 2003.

[9] A. Seddik-Ghaleb, Y. Ghamri-Doudane, and S. M. Senouci, *"TCP WELCOME TCP Variant for Wireless Environment, Link losses, and Congestion packet loss ModEls," in First International Communication Systems and Networks and Workshops, COMSNETS* 2009.

[10] Dongkyun Kim, Hanseok Bae, Jeomki Song, *"Analysis of the Interaction between TCP Variants and Routing Protocols in MANETs", IEEE ICPP 2005 Workshops. International Conference Workshops*, South Korea ,14-17 June 2005, Pp. 380-386.

[11] Ikeda, M., *"Performance Evaluation of AODV Protocol for Single and Multiple Traffic in MANETs Considering Packet Delivery Fraction Parameter", Emerging Intelligent Data and Web Technologies (EIDWT), 2012 Third International Conference*, 19-21 Sept. 2012, Pp. 74-80.

[12] Khuzairi Mohd Zaini, Adib M. Monzer Habbal *"An Interaction between Congestion-Control Based Transport Protocols and Manet Routing Protocols", Journal of Computer Science , Malaysia,* 2012, Pp. 468-473.

[13] Shitalkumar Jain, Shrikant Kokate, Pranita Thakur, Shubhangi Takalkar Maharashtra, *"A Study of Congestion Aware Adaptive Routing Protocols in MANET", Computer Engineering and Intelligent Systems, Vol 3, No.4, Maharashtra Academy of engineering,* Alandi, 2012.

[14] Sujata V. Mallapur, Siddarama . R. Patil, *"Survey on Simulation Tools for Mobile Ad-Hoc Networks", IRACST – International Journal of Computer Networks and Wireless Communications (IJCNWC)*, Vol.2, No.2, April 2012.

[15] Christian Lochert Bjorn Scheuermann Martin Mauve , *"A Survey on Congestion Control for Mobile Ad-Hoc Networks", Computer Science Department, University atsstr.* D-40225 Dusseldorf, Germany.

[16] Sreenivas B.C, G.C. Bhanu Prakash, K.V. Ramakrishnan, *"L2DB-TCP: An adaptive congestion control technique for MANET based on link layer measurements", Advance Computing Conference (IACC), 2013 IEEE 3rd International*, 22-23 Feb. 2013.

[17] Das, M., Sahu, B., *"Analysis of effect of mobility on performance of AODV in Mobile Ad hoc Network", Computer Communication and Informatics (ICCCI), 2012 International Conference, Coimbatore*, 10-12 Jan. 2012.

[18] Aleksandar Milenkoski , Biljana Stojcevska, *"Loss Differentiation algorithms vs. Congestion Control Schemes: Dynamics and Performance", International Journal of Distributed and Parallel systems (IJDPS)* Vol.1, No.1, September 2010, Pp. 13-30.