Ravinder Kaur, Gurpreet Singh Josan / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 5, September- October 2012, pp.1682-1688

Performance Evaluation Of Congestion Control Tcp Variants In Vanet Using Omnet++

Ravinder Kaur*, Gurpreet Singh Josan**
*(Department of Computer Science, Punjabi University, Patiala)
** (Department of Computer Science, Punjabi University, Patiala)

ABSTRACT

A Vehicular Ad-hoc network or VANET is a technology that uses moving cars as node in a network to create a mobile network. A Vehicular Ad-hoc Networks (VANET) is an area of wireless technologies that are attracting a great deal of interest. There are still several areas of VANETS, such as medium access control, security, routing protocols, congestion control that lack large amounts of research. There is also a lack of freely available simulators that can quickly and accurately simulate VANETs. The paper aims to investigate the performance of Congestion Control TCP Variants in VANET by using routing protocols i.e. AODV (Adhoc on Demand Distance Vector) and DSR(Dynamic Source Routing) . Delay and Throughput are the two parameters that are consider to grade the TCP Variants. Conclusions are drawn based on the evaluation results using OMNET++ and SUMO simulator. The results clearly show that New Reno is better than that of Reno but the performance with Tahoe is as that with New Reno Variant except that in a large network size Tahoe achieves less Delay and better Throughput. Furthermore, it can be observed that New Reno is better than that of TCP Reno but cannot be as good as that of Tahoe.

Keywords - Ad-hoc Networks, TCP variants, Routing Protocols, AODV, DSR.

1. INDRODUCTION

VANET is a technology that uses moving cars as nodes in a network to create a mobile network [1] .It turns every participating car into a wireless router or node, allowing cars approximately 100 to 300 meters of each other to connect and, in turn, create a network with a wide range. Mobile ad hoc Networks (MANETs) are mainly linked with mobile laptops or wireless handheld devices, whereas VANET is concerned with vehicles (such as cars, vans, trucks, etc). Mobile adhoc networks (MANETs) are a type of wireless network that does not require any complicated infrastructure. But in case of VANET technology each moving cars is consider as nodes in a network to create a mobile network with a wide range in which cars fall out of the signal range and drop out of the network, other cars can join in, connecting vehicles to one another so that a Mobile Internet is created [2]. And this technology will also integrated with police so that fire vehicles can communicate with police for safety purpose. Other purposes include essential alerts and accessing comforts and entertainment used.

VANET bring new challenges to design an efficient routing protocol for routing data among vehicles, called V2V or vehicle to vehicle communication. As cars fall out of the signal range and drop out of the network, other cars can join in, connecting vehicles to one another so that a Mobile Internet is created. It is estimated that the first systems that will integrate this technology are police and fire vehicles to communicate with each other for safety purposes [3]. Other purposes include essential alerts and accessing comforts and entertainment. VANETs are a kind of MANETs provide vehicle to vehicle (V2V) and vehicle to roadside wireless communications, this means that every node can move freely within n/w coverage and stay connected. Vehicles are equipped with wireless transceivers and computerized control modules are used. [4] in network to avoid accidents etc. and In this context, we evaluate the performance of TCP Variants (TCP Reno, TCP new Reno, TCP Tahoe) using routing protocols AODV and DSR on basis of parameter throughput and delay which can perform efficiently with increasing number of vehicles in a network by developing coupling between OMNet++ (network simulator) and SUMO (traffic simulator) by using Traci as a interface [5][6][7].

2. TCP VARIANTS

TCP is transport layer is the reliable connection orientated protocol that provides reliable transfer of data between the nodes. It ensures that the data is reached the destination correctly without any loss or damage. The data is transmitted in the form of continuous stream of octets. The reliable transfer of octets is achieved through the use of a sequence number to each octet. Another aspect of TCP is the tree way handshakes mechanism to establish a connection between the nodes [8]. Furthermore, TCP uses the port assignment as an addressing mechanism to differentiate each connection for the cases of more TCP connection between nodes are required. After the introduction of first version of TCP several different TCP variants exist. The most famous implementation of TCP called Tahoe, Reno and New-Reno.

2.1 Overview of Congestion Control TCP Variants

Modern TCP implementations contain a number of algorithms aimed at controlling network congestion while maintaining good user throughput. Early TCP implementations followed a go-back- model using cumulative positive acknowledgment and requiring a retransmit timer expiration to re-send data lost during transport. So modern TCP implementations lead to minimize network congestion. The three TCP Variants that we are using are discussed below:

I) **TCP Tahoe** *:*

TCP Tahoe was released in 1998. TCP Tahoe (1989) release has the following features: slow start, congestion avoidance and fast retransmit. The idea of TCP Tahoe is to start the congestion window at the size of a single segment and send it when a connection is established. If the acknowledgement arrives before the retransmission timer expires, add one segment to the congestion window. This is a multiplicative increase algorithm and the window size increases exponentially[11]. The window continues to increase exponentially until it reaches the threshold that has been set. This is the Slow Start Phase. Once the congestion window reaches the threshold, TCP slows down and the congestion avoidance algorithm takes over. Instead of adding a new segment to the congestion window every time an acknowledgement arrives, TCP increases the congestion window by one segment for each round trip time. This is an additive increase algorithm. To estimate a round trip time, the TCP code uses the time to send and receive acknowledgements for the data in one window. TCP does not wait for an entire window of data to be sent and acknowledged before increasing the congestion window. Instead, it adds a small increment to the congestion window each time an acknowledgement arrives. The small increment is chosen to make the increase averages approximately one segment over an entire window. When a segment loss is detected through timeouts, there is a strong indication of congestion in the network. The slow start threshold is set to one-half of the current window size. Moreover, the congestion window is set to 1 segment, which forces slow start[10].

II) TCP Reno

This Reno retains the basic principle of Tahoe, such as slow starts and the coarse grain re-transmit timer. However it adds some intelligence over it so that lost packets are detected earlier and the pipeline is not emptied every time a packet is lost [11] Reno requires that we receive immediate acknowledgement whenever a segment is received. The logic behind this is that whenever we receive a duplicate acknowledgment, then his duplicate acknowledgment could have been received if the next segment in sequence expected, has been delayed in the network and the segments reached there out of order or else that the packet is lost. If we receive a number of duplicate acknowledgements then that means that sufficient time have passed and even if the segment had taken a longer path, it should have gotten to the receiver by now[10]. There is a very high probability that it was lost. So Reno suggests an algorithm called **'Fast Re-Transmit'.**

III) New Reno

New RENO is a slight modification over TCP-RENO. It is able to detect multiple packet losses and thus is much more efficient that RENO in the event of multiple packet losses. Like Reno, New-Reno also enters into fast-retransmit when it receives multiple duplicate packets, however it differs from RENO in that it doesn't exit fast-recovery until all the data which was out standing at the time it entered fast recovery is acknowledged. Thus it overcomes the problem faced by Reno of reducing the CWD multiples times. The fast-transmit phase is the same as in Reno. The difference in the fast recovery phase which allows for multiple re-transmissions in new-Reno. Whenever new-Reno enters fast recovery it notes the maximums segment which is outstanding. The fast-recovery phase proceeds as in Reno, however when a fresh ACK is received then there are two cases: If it ACK's all the segments which were outstanding when we entered fast recovery then it exits fast recovery and sets CWD to ssthresh and continues congestion avoidance like Tahoe. If the ACK is a partial ACK then it deduces that the next segment in line was lost and it re-transmits that segment and sets the number of duplicate ACKS received to zero. It exits Fast recovery when all the data in the window is acknowledged **[12][20]**.

2.2 Routing protocols

As we are comparing TCP Variants on the basis of Routing Protocols AODV and DSR as discussed below:

I) AODV

The Ad-hoc On Demand Distance Vector (AODV) is considered an efficient VANET routing protocol. The AODV routing protocol utilizes an on-demand technique in order to discover the routes. This means that the route between two endpoints (nodes) is formed as per requirement for the source node and maintained as long as the routes are needed. Moreover, the protocol uses a destination sequence number to recognize the most recent path and to guarantee the freshness of the routes. Reactive protocols like AODV shrinks the control traffic overhead at the cost of higher latency in discovering new routes [13].

II) DSR

Dynamic Source Routing (DSR) is a widely used reactive (on-demand) routing protocol which is designed for mobile ad-hoc networks. DSR permits the network to run without any existing network infrastructure and thus the network becomes as a self-organized and self-configured network. This protocol maintains an on-demand approach and hence extinguishes the periodic table-update messages needed in the table-driven approach [13].

3. SIMULATION AND ENVIRONMENT

In simulation the different types of scenarios are consider based upon traffic density. In this paper a comparison between different TCP variants are made based upon routing protocol on wireless network of City, Country etc. The investigation involves the measurement of delay and throughput of the network in each of the above cases. Finally, the results achieved for each case of TCP variants with different routing protocols, number of nodes in the networks will be assessed and then summarized result is evaluated based upon those result.

3.1 Throughput

Throughput is the ratio of total amount packets the receiver will receive from the source of the data within the specified time frame. End to end delay for the packet transmission is most important metrics for the throughput performance of the routing protocols. Along with the routing protocols in the wireless networks for the performance analysis the routing agents are also needs to consider congestion control TCP agents such as TCPTahoe , TCPReno and TCP New Reno In this paper, AODV and DSR protocols are simulated with different TCP agents such as TCP New Reno, TCP Reno, TCP Tahoe for the different number of mobile nodes and networks sizes. We measured the throughput of every scenario as shown in the Table3.1, Table3.2 and Table3.3. that are showing the average throughput performance for AODV and DSR with TCP-Reno, TCP-NewReno, TCPTahoe. Based on these readings we prepared following performance comparison graphs for throughput performance. Following are the graphs from Fig 3.1 to Fig 3.6 for each scenario with different routing protocols and different TCP agents. Here measurement of the throughput is calculated by calculating the throughput of receiving the packets versus total simulation.

Table3.1 High Traffic Density- Throughput (bits/sec)

Protocols	TCP VARIANTS	City	Country	Highway
AODV	TCP RENO	202.8	296.97	205.7
DSR	TCP RENO	262.8	302.95	245.58
AODV	TCP NEW RENO	241.39	316.43	216.18
DSR	TCP NEW RENO	302.03	334.27	275.48
AODV	TCP TAHOE	279.05	320.95	247.6
DSR	TCP TAHOE	321.49	342.72	289.5

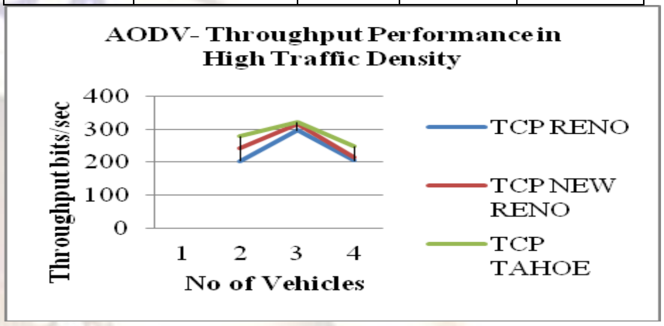

Fig 3.1: AODV-TCP variants Throughput performance in High Traffic density

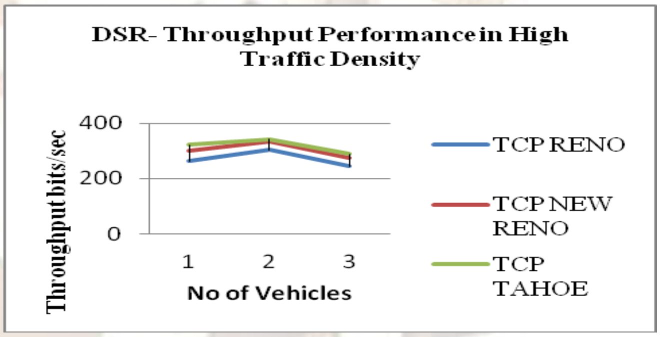

Fig 3.2: DSR-TCP variants throughput performance in High Traffic density

Table3.2 Medium Traffic Density- Throughput (bits/sec)

Protocols	TCP VARIANTS	City	Country	Highway
AODV	TCP RENO	402.78	375.78	307.85
DSR	TCP RENO	365.34	354.89	209.65
AODV	TCP NEW RENO	305.79	389.62	340.51
DSR	TCP NEW RENO	424.8	402.63	321.72
AODV	TCP TAHOE	365.33	245.67	320.95
DSR	TCP TAHOE	234.76	375.98	279.12

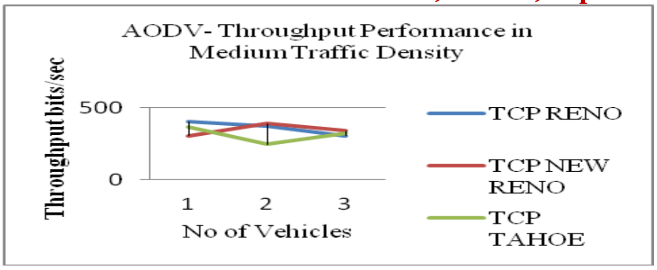

Fig 3.3: AODV-TCP Variants Throughput performance in Medium Traffic density

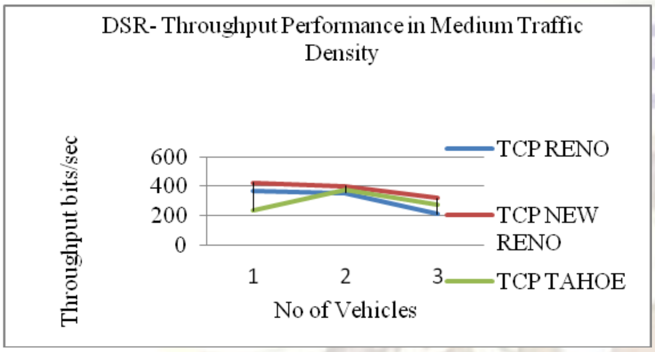

Fig 3.4: DSR-TCP variants Throughput performance in Medium Traffic density

Table 3.3 Low Traffic Density- Throughput (bits/sec)				
Protocols	TCP VARIANTS	City	Country	Highway
AODV	TCP RENO	486.78	424.56	507.86
DSR	TCP RENO	400.54	300.65	243.76
AODV	TCP NEW RENO	543.65	455.67	396.76
DSR	TCP NEW RENO	576.32	410.45	305.78
AODV	TCP TAHOE	456.76	346.98	415.25
DSR	TCP TAHOE	342.98	456.98	351.43

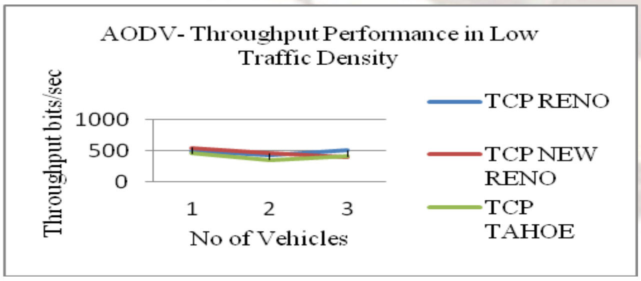

Fig 3.5: AODV-TCP variant Throughput performance in Low Traffic density

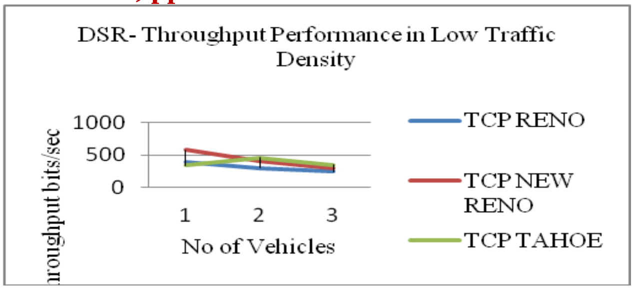

Fig 3.6: DSR-TCP variants Throughput performance in Low Traffic density

3.2 Delay

This one more performance metrics which we calculated here for all the TCP variants with the both routing protocols AODV and DSR with different network scenarios. Following Tables 3.4, Table 3.5 and Table 3.6 shows the average end to end delay performance for AODV and DSR with TCP-Reno, TCP-New Reno and TCP-Tahoe which will explain the performance effects of TCP variants with AODV and DSR network routing protocols and from Fig 3.7 to Fig 3.12 shows a Graph for delay performance for each scenario with different routing protocols and different TCP agents.

Table 3.4 High Traffic Density- Delay/sec				
Protocols	TCP VARIANTS	City	Country	High way
AODV	TCP RENO	498.23	281.56	373.41
DSR	TCP RENO	477.1	202.58	204.12
AODV	TCP NEW RENO	98.94	77.6	104.61
DSR	TCP NEW RENO	87.98	42.03	34.44
AODV	TCP TAHOE	97.5	73.97	102.72
DSR	TCP TAHOE	86.17	41.91	32.45

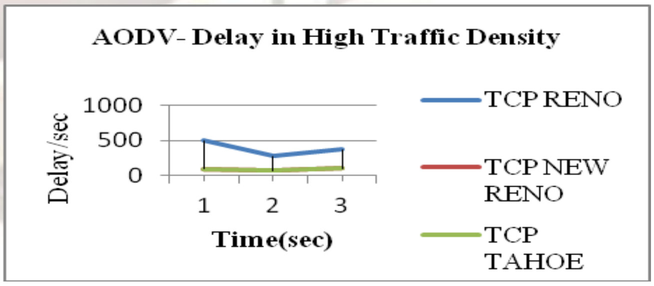

Fig 3.7: AODV-TCP variants Delay performance in High Traffic density

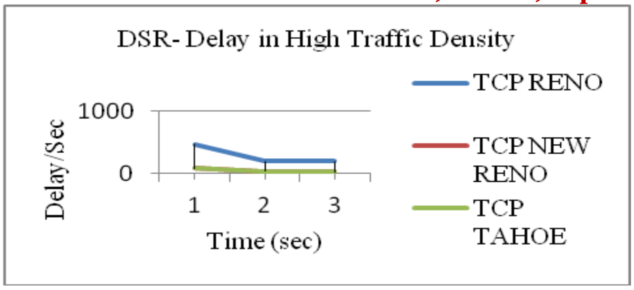

Fig 3.8: DSR-TCP variants Delay performance in High Traffic density

Table3.5 Medium Traffic Density-Delay/sec

Protocols	TCP VARIANTS	City	Country	Highway
AODV	TCP RENO	515.3	285.56	387.67
DSR	TCP RENO	495.1	205.34	215.21
AODV	TCP NEW RENO	85.9	78.56	117.18
DSR	TCP NEW RENO	74.78	45.78	45.23
AODV	TCP TAHOE	96.54	75.89	120.24
DSR	TCP TAHOE	85.17	47.89	39.46

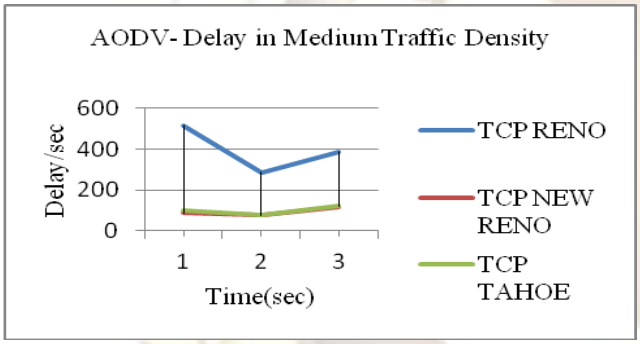

Fig 3.9: AODV-TCP variants Delay performance in Medium Traffic density

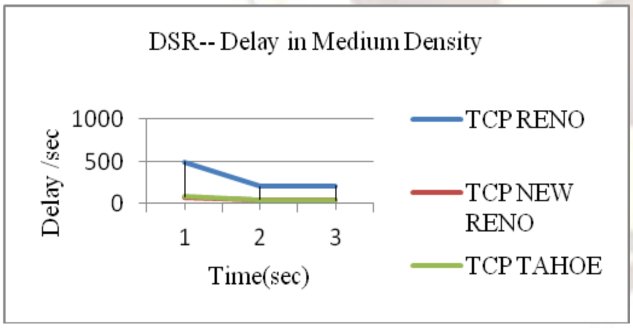

Fig 3.10: DSR-TCP variants Delay performance in Medium Traffic density

Table3.6 Low Traffic Density-Delay/sec

Protocols	TCP VARIANTS	City	Country	Highway
AODV	TCP RENO	498.23	281.56	373.41
DSR	TCP RENO	477.1	202.58	204.12
AODV	TCP NEW RENO	45.61	39.15	78.27
DSR	TCP NEW RENO	77.27	24.17	32.45
AODV	TCP TAHOE	97.5	73.97	102.72
DSR	TCP TAHOE	86.17	41.91	32.45

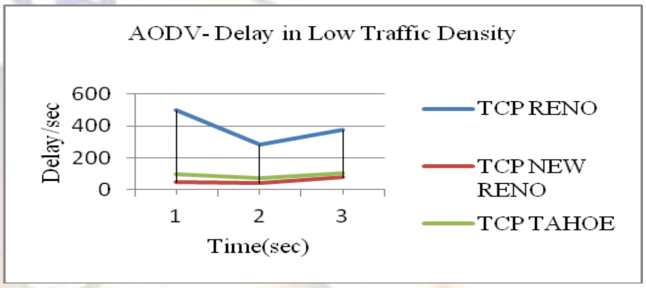

Fig 3.11: AODV-TCP variants Delay performance in Low Traffic density

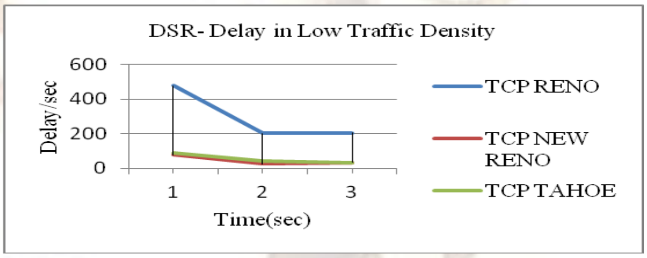

Fig 3.12: DSR-TCP variants Delay performance in Low Traffic density

Based upon above tables and graphs the summarized tables have been created for TCP Tahoe, TCP Reno and New Reno as shown in Table 3.7, Table 3.8, Table 3.9 and results are concluded on the basis of that.

Table3.7 TCP TAHOE Summary Table

Protocol	Traffic density	Max Delay	Min Delay	Max Throughput	Min Throughput
AODV	High	102.72	73.97	320.95	247.6
DSR	High	86.17	32.45	342.72	289.6
AODV	Medium	120.24	95.89	365.33	245.67
DSR	Medium	85.17	39.46	375.98	234.76
AODV	Low	102.72	93.97	456.76	346.98
DSR	Low	86.17	32.45	456.98	342.98

Summarized result of TCP Tahoe with AODV and DSR

From the table it is clear that AODV has less Delay for small scale network whereas the network size increases DSR become less delay as compared to AODV. From the throughput matter is opposite, AODV achieve better throughput as the network size increased in TCP Tahoe.

Table3.8 TCP NEW RENO Summary Table

Protocol	Traffic density	Max Delay	Min Delay	Max Throughput	Min Throughput
AODV	High	104.61	77.6	316.43	216.18
DSR	High	87.98	34.44	334.27	275.48
AODV	Medium	117.18	78.56	389.62	305
DSR	Medium	74.78	45.23	424.8	321.72
AODV	Low	78.27	39.15	543.65	396.76
DSR	Low	77.27	24.17	576.32	305.78

Summarized Result of TCP New Reno with AODV and DSR

It is clear from the table that AODV has less Delay for large scale network whereas the network size decreases the DSR become less delay as compared to AODV and in case of Throughput the matter is opposite DSR achieve better Throughput in small network whereas AODV achieve better in case of large network.

Table 3.9 TCP RENO Summarized Table

Protocol	Traffic density	Max Delay	Min Delay	Max Throughput	Min Throughput
AODV	High	498.23	281.56	296.97	202.8
DSR	High	477.1	202.58	302.95	245.58
AODV	Medium	515.3	285.56	402.78	307.85
DSR	Medium	495.1	205.34	365.34	209.65
AODV	Low	498.23	281.56	507.86	424.56
DSR	Low	477.1	202.58	400.54	243.76

Summarized Result of TCP Reno with AODV and DSR

It is clear from the table that DSR have less Delay for small scale network whereas AODV have less Delay for large scale network as compared to DSR and in case of Throughput AODV achieve better Throughput for small size network, whereas DSR achieve better Throughput as network size increases. Now, we conclude that AODV Protocol achieve better performance as compared to DSR protocol from the throughput point of view .The situation is different when considering the Delay as a performance parameter

Result

It can be observed that New Reno is better than that of Reno but the performance with Tahoe is as that with New Reno Variant except that in a large network size Tahoe achieves less Delay and better Throughput. Furthermore, it can be observed that New Reno is better than that of TCP Reno but cannot be as good as that of Tahoe.

4. Conclusion and Future Work

4.1 Conclusion

The main purpose of this paper, to analyze the performance of the three most widely used TCP variants (Reno, New Reno and TAHOE) in an ad-hoc environment with respect to the two protocols i.e. AODV and DSR and to know how well these variants respond to different network conditions, particularly with respect to extension of network size .In this paper we discuss how the different mechanism affect the through put and Delay of TCP Variants. we conclude that AODV Protocol achieve better performance as compared to DSR protocol from the throughput point of view .The situation is different when considering the Delay as a performance parameter and it can also be observed that the performance with Tahoe is as that with New Reno Variant except that in a large network size Tahoe achieve less Delay and better Throughput. Furthermore, it can be observed that New Reno is better than that of TCP Reno but cannot be as good as that of Tahoe.

4.2 Future Work

As we, selected these numerous TCP congestion control protocols of interest by simulation in an OMNET++ tool, another possibility of doing the same work can be done through another tool like NS-3, Qualnet. Also, selection of other congestion control protocols can be use for the performance evaluation or other parameters of performance could be considered for simulation.

References

[1] H. Hartenstein and K.P. Laberteaux, "A Tutorial Survey on Vehicular Ad Hoc Networks," IEEE Comm. Magazine, vol. 46, no. 6, pp. 164-171, June 2008.

[2] A.K. Saha and D.B. Johnson, "Modeling Mobility for Vehicular Ad-Hoc Networks," Proc. First ACM Int'l Workshop Vehicular

Ad Hoc Networks (VANET '04), pp. 91-92, Oct. 2004.

[3] H. Fubler, M. Kasemann, and D. Vollmer,"A comparison of strategies for vehicular ad-hoc networks," Dept. of Comp. Sc., Univ. of Mannheim, Tech. Rep. TR-3-2002, 2002.

[4] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," Wireless Comm. and Mobile Computing, special issue on mobile ad hoc networking: research, trends and applications, vol. 2, no. 5, pp. 483-502, 2002.

[5] A.Varga, "The OMNeT++ Discrete Event Simulation System," Proc. European Simulation Multiconf. (ESM '01), June 2001.

[6] D.Krajzewicz et al, "SUMO (Simulation of Urban MObility); An Open-Source Traffic Simulation," Proc. Fourth Middle East Symp. Simulation and Modelling (MESM '02), pp. 183-187, Sept. 2002.

[7] S. Fischer et al, "TRACI: An Interface for Coupling Road Traffic and Network Simulators," *Proceedings of the 11th Communications and Networking Simulation Symposium*, pp. 155–163, 2008.

[8] S. Floyd and K. Fall, "Simulation based comparisons of Tahoe, Reno, and SACK TCP," *ACM Computer Communication Review*, vol. 26, no. 3, pp. 5–21, July 1996.

[9] H.Balakrishnan et al, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," *SIGCOMM Symposium on Communications Architectures and Protocols*, Aug. 1996.

[10] F. Anjum and L. Tassiulas, "Comparative study of various TCP versions over a wireless link with correlated losses," *IEEE/ACM Transactions on Networking*, vol. 11, no. 3, pp. 370–383, June 2003.

[11] Laxmi .S et al," Performance Evaluation of TCP Tahoe, Reno, Reno with SACK, and New Reno Using OPNET Modeler", Simon Fraser University Vancouver, British Columbia Canada,2004.

[12] S. Floyd and T. Henderson, "The New Reno modification to TCP's fast recovery algorithm," *RFC 2582*, Apr. 1999.

[13] C.Sommer et al , "Simulation of Ad Hoc Routing Protocols using OMNeT++," *Mobile Networks and Applications*, pp. 786–801, Jun. 2009.

[14] A.Mahajan et al , "Urban Mobility Models for VANETs," Proc. Second IEEE Int'l Workshop Next Generation Wireless Networks (IEEE WoNGeN '06), Dec. 2006.

[15] H. Lee, S. Lee, and Y. Choi, "The influence of the large bandwidth-delay product on TCP Reno, New Reno, and SACK," in *Proc. Information Networking Conference*, Oita, Japan, Feb. 2001, pp. 327–334.

[16] D.Borman, R. Braden, and V. Jacobson, "TCP Extensions for High Performance," Request for Comments (Proposed Standard) RFC 1323, Internet Engineering Task Force, May 1992. (Obsoletes RFC1185).

[17] H. Lee, S. Lee, and Y. Choi, "The influence of the large bandwidth-delay product on TCP Reno, New Reno, and SACK," in *Proc. Information Networking Conference*, Oita, Japan, Feb. 2001, pp. 327–334.

[18] M. Piorkowski et al, "Joint Traffic and Network Simulator for VANETs," Proc. Mobile Information and Comm. Systems (MICS '06), Poster Session, Oct. 2006.

[19] R. Paul and Lj. Trajkovic, "Selective-TCP for wired/wireless networks," in *Proc. SPECTS 2006*, Calgary, AL, Canada, Aug. 2006, pp. 339–346.

[20] Simulation of urban mobility," Jan. 2011. [Online]. Available: http://sumo.sourceforge.net/.

[21] " Open street maps: free editable map of the whole world," Jan. 2011. [Online]. Available: http://www.openstreetmap.org.

[22] "OMNeT++", Available: http://www.omnetpp.org.