J. Engg. and Appl. Sci. Vol. 32 No. 2 July - December 2013 *ISSN 1023-862X*

EVALUATING PERFORMANCE OF DIFFERENT ROUTING PROTOCOLS IN MANET ENVIRONMENT BASED ON TCP WINDOW SIZE EVALUATION

Zeeshan Sabir[*], M. Inayatullah Babar[*], Saeed Ur Rehman[**], Amjad Ullah Khattak [*] Syed Riaz Ul Hassnain[*]

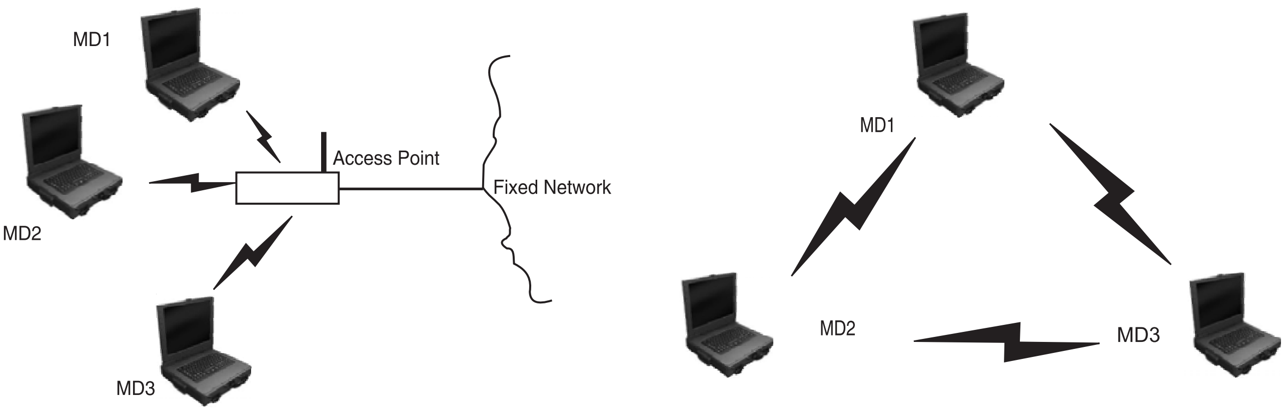
ABSTRACT

Mobile Adhoc Networks (MANET) refers to an arrangement of wireless mobile nodes that have the tendency of dynamically and freely self-organizing into temporary and arbitrary network topologies. Performance of the MANETs primarily depends upon the protocol running at the backend for routing the packets between the mobile nodes of the network. Each of these protocols has its own pros and cons in different scenarios. In this paper we have simulated an environment of MANETs consisting of a few Mobile nodes that tend to form a network amongst eachother during their random mobile behaviour. The environment has been simulated using three different routing protocols via the Network Simulator(NS)-2.35 scripting and the performance of these protocols have been compared based upon their TCP window size evaluation and the number of tcp packets successfully received at the receiving node under each scenario.

KEY WORDS: *MANET, DSR, AODV, DSDV, TORA, NS-2.34, TCP, CBR, Buffer Management, Congestion Window, Access Point (AP), Reactive/Proactive Protocol.*

INTRODUCTION

MANET is a popular wireless network architecture that has proven to be easily deployable in different environments without the support of any fixed infrastructure. More specifically, MANET represents a set of wireless mobile nodes that have the tendency of communication with the other nodes using an adhoc network without needing any pre-existing backbone hardware[1]. Depending upon whether a centralized entity is controlling the routing of packets in the network, MANETs are broadly classified into two basic groups. Infrastructure-based and infrastructure-less Wireless Adhoc Networks. The centralized node that is used to control the routing of the packets in the network is termed as the Access Point (AP). Presence of AP makes the MANET an Infrastructure-based wireless Adhoc Network. A schematic comparison of both these types of Wireless Adhoc Networks is shown in Figure. 1.



a. Infrastructure-based Wireless Mobile Adhoc Network

b. Infrastructure-less Wireless Mobile Adhoc Network

Figure. 1: Schematic Comparison of Infrastructure-based and Infrastructure-less Wireless Adhoc Network

[*]University of Engineering & Technology, Peshawar, Pakistan
[**]The University of Auckland, New Zealand

Depending upon the routing protocol implemented at the Network Layer, MANETs show immense adaptivity and auto-configurability in various environments which makes them suitable for flexible deployments in different scenarios including emergency situations, rescue services, military insurgencies, industrial applications and campus-wide networks[2,3].

There are a number of different protocols that are used for routing the packets amongst the nodes of MANETs. Some of the widely used standardised protocols in MANETs are the Dynamic Source Routing (DSR) Protocol, Adhoc On-Demand Distance-Vector (AODV) Routing Protocol and Destination Sequenced Distance Vector (DSDV) Routing Algorithm. All these algorithms have different metrics for choosing a particular route to the destination. A brief introduction about the path selection metric and routing path formation for each of these protocols is given in next section.

BRIEF INTRODUCTION TO ROUTING PROTOCOLS DSR

DSR algorithm is designed for upto 200 nodes[4] and works in a reactive manner i.e. the Route-Request message are send in a broadcast way to all the neighbours whenever the node has data to send[5]. The receiving node looks for the destination address in their cache memory. If they find it they send the reply by piggybacking it on the route-request message. In this way the route request is entertained in DSR. Similarly the Route maintenance is carried-out in DSR by confirmation of link availability for carrying data using an acknowledgement packet by the receiving node.

AODV

As mentioned from the name, AODV uses a reactive approach and constructs the routing paths on-demand[6]. Using a destination sequence number for each of the route entry in its routing table it uses three control messages for maintaining and discovering the links[7]. These messages include Route Error (RErr), Route Requests (RReqs) and Route Reply (RRep). A Hello message is broadcasted to get information about the one-hop neighbours. RErr is send to the initiating Source node whenever the destination route is either gone or is not accessible via the current intermediate node. So depending upon the current need of the route, source node reinitiates the RReq and a route is specified for this particular communication session.

DSDV

Being a comparatively early protocol, DSDV is suitable for connecting small number of nodes through ad-hoc network[8]. Similar to the other proactive protocols, DSDV utilizes its battery power inefficiently by constantly updating its routing paths, even when the node is in the idle mode. Inspired by the Bellman-Ford algorithm, each routing table entry in the DSDV protocol utilizes a sequence number generated by the destination. This sequence number is a measure of the freshness of the route. In order to solve the Routing loop problem, the sender sends the update with this sequence number. Priority of the route selection is based upon the sequence number and route with high sequence number is preferred over the others for selection.

RELATED WORK

Some work has already been done in the field of performance evaluation and comparison of the MANET protocols. Good comparison of the AODV, DSR and TORA routing protocols in MANETs has been given by Anvj et al[9]. In this work the authors have simulated an environment of MANET using Random Waypoint model in NS-2, in which the nodes are free to move within a transmission range of 250 m. Constant Bit Rate (CBR) traffic is used amongst the nodes of the network for transferring data. Results have been shown for throughput and end-to-end delay for all the three algorithms and are compared in a very good manner. The two deficiencies in the results are the lack of the Window Size evaluation as-per the movement pattern of the nodes and secondly the traffic that is considered is CBR while in actual practice most of the traffic consists of variable bit rate traffic and TCP-based. So some further effort in this work might have produced better and more realistic results.

Another very good effort in the field of analysing the TCP performance based on congestion window size in MANET environment is given in Kim et al[10]. In this work a sort of adaptive approach is proposed for estimating the optimum size of the congestion window. This optimum size is then advertised to the rest of the network as extra control information. Every node then sets the size of its congestion window to this value of the advertised window for improved performance of the system. This work reveals a much improved performance of the MANET based environment but at the expense of extra computational burden on each node (for infrastructure-less system) or on the AP (for infrastructure-based system). Similarly the advertisement of the adaptive congestion window size also puts an extra control information introduction into the system.

Mbarushimana *et.al*[11] has done a comparative study of different active (OLSR) and reactive(AODV, DSR) routing protocols in MANETs using OPNET simulator. They have simulated an environment of MANET and have shown that the proactive protocols outperform the reactive protocols at the expense of large system overhead due to flooding of regular routing table updates for keeping the routing table fresh. Results have been shown for throughput, end-to-end delay and routing traffic load per node. Again the use of a Constant-Bit-Rate traffic tilt the results far from the real-time since in real-time cases the traffic is variable bit rate and can cause more packet drops at the nodes then did the CBR traffic.

In other eminent works[12,13,14] the TCP performance for MANET has been evaluated in the environment of moving nodes. TCP-Reno has been considered as the basic algorithm governing the transmission of packets in the network. Results have been shown for TCP Throughput vs node's speed and TCP Throughput vs Number of Hops Count etc. Another emphasis of the referred study is the number of link failures as the mobility of the nodes in the simulated scenario increases. The referred studies prove to be a very fine addition in the field of the analysis of the TCP performance in MANETs. But the discussion of congestion window in the context of node's movement is missing in these works.

Samir *et.al* [15], has used a discrete-event, packet level Network Routing Simulator called MaRS (Maryland Routing Simulator)[16] to show that the Proactive protocols tend to show least packet drop rate but at the cost of high system routing overhead. On the other hand the reactive protocols are bandwidth efficient but show more packet dropage ratio and end-to-end delay due to the loss of information regarding distance. This study is conducted in a limited scenario using CBR traffic and taking into account the ideal condition that all the active links of the network are using a separate frequency band. Also size of the network is kept much moderate or small. Results have not been shown for big size network. The size of the window during the whole course of movement of nodes is not at all discussed in the results portion.

In this paper we have used Network Simulator(NS)-2.35 to simulate and evaluate a MANET environment using the three most commonly used proactive and reactive routing protocols. NS-2.35, an Open-Source Network Simulator has been used for its efficiency of simulating different environments of MANETs using various network size, load and nodes speed. The basic emphasis has been given to the TCP window size evaluation for showing the impact of the nodes movement when a connection is made or broken during the course of movement of the nodes in the network. All the nodes are kept mobile during the course of trace statistics collection. Results have also been shown for the successfully received tcp packets at the receiving node for different movement scenarios and conditions using these protocols.

SIMULATED ENVIRONMENT AND RESULTS

We have used Open-Source Network Simulator (NS)-2.35 over Fedora-14 platform for simulating the MANETs environment using different Routing algorithms. The environment has been simulated defining three nodes moving in an area of 800m x 500m. The simulation setup is given in Table 1.

In the start of simulation, the nodes are much far-apart and there is no connection between the nodes. The initial position of the nodes is shown in the NAM screenshot of Figure 2.

Being origin of the screen axis at the lower-right corner, at 10, 15 and 20 seconds, node 0, 1 and 2 starts their movement respectively. Node 0 starts its movement in the down-left direction with 1 m/s while node 1 starts moving towards its right with 3 m/s while node 2 to its down-left with 1 m/s. The actual NS-2.35 script dictating the movement scenario of the nodes in the simulation is given as under:

Table I. Simulation setup for different protocols

Routing Protocols	DSDV, DSR, AODV
MAC Layer	IEEE 802.11 for Wireless LAN
Bandwidth	11 Mbps
Data Traffic	FTP (over TCP)
Simulation Time	280 seconds
Terrain	800m x 500m
Nodes Placement	Random
Mobility Scenario	Random Waypoint Mobility Model
No. Of Nodes	3
Queue Type	Queue/DropTail/PriQueue, CMUPriQueue

movement generation of the nodes

$ns at 10.0 “$node_(0) setdest 250.0 250.0 1.0”

$ns at 15.0 “$node_(1) setdest 45.0 285.0 3.0”

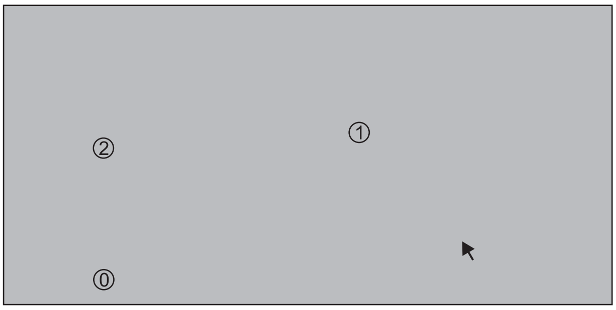


Figure. 2: NAM Screenshot Showing Initial Positions of the Three Nodes

$ns at 20.0 “$node_(2) setdest 310.0 30.0 1.0”

$ns at 90.0 “$node_(2) setdest 200.0 280.0 1.5”

$ns at 110.0 “$node_(0) setdest 480.0 300.0 2.0”

$ns at 185.0 “$node_(0) setdest 480.0 300.0 4.5”

An FTP agent is defined at the node 0 which uses TCP traffic for data transmission. While Null agent is defined at node 1. When the first tcp packet is transmitted at 10 seconds, there is no connection established between the two nodes. In the meanwhile the three nodes start their movement towards their destination. At the first timeout that occurs 6 seconds later i.e at the 16th second since the simulation started, another transmission reattempt occurs but still no connection can be established between the two nodes. In the meanwhile the nodes continue to move closer and closer during there random movement, the next reattempts occur at 28th, 52nd and then at around 100th second. The two nodes are now in close vicinity of each other through the node 2 thus a two hop path is established between the two nodes at about 96th sec since the start of the simulation. This is depicted in Figure 3.

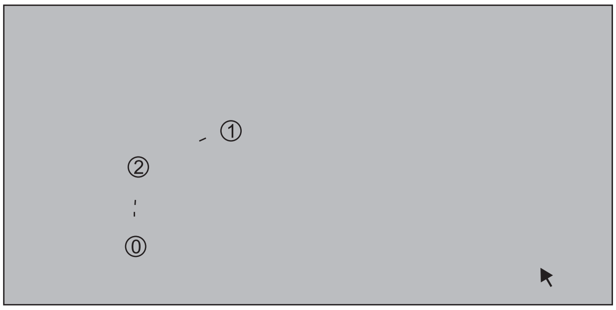


Figure. 3: Two-Hop path Formation Between the Three Nodes at 96.2 Sec

As this event occurs, the two nodes start transmitting packets to each other. Data is being sent from Node 0 to Node 1 while Ack packets are sent from Node 1 to Node 0. During the phase of the connection, the two nodes became so close to each other that a direct path is established between the two at around 110 sec since the start of the simulation. This is depicted in Figure 4.

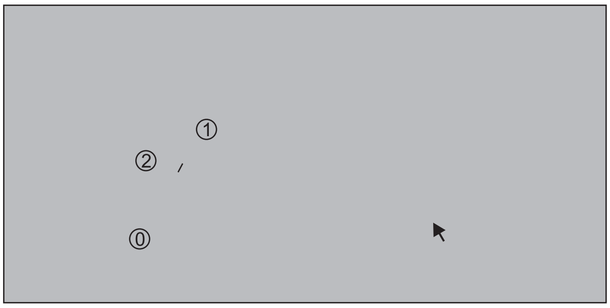


Figure. 4: Direct Path Establishment Between Nodes 0 and 1 at 110.3 Sec

During the phase of handover of the connection from 2-hop to 1-hop, there is a slight phase of packet-drop which is shown by the window size drop at around 110 secs after which it again starts rising till 200 seconds when the two nodes again go out of range and a connection can not be established between the two. This is again shown by a packet drop between the two nodes as shown in Figure. 5.

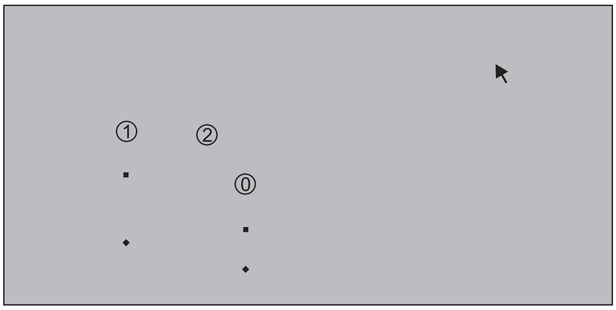


Figure 5: Loss of packets by the two nodes due to a direct Path Loss When the Nodes Get Out-of-Range at 202.3 Secs.

During the random course of movement of the three nodes, at around 235 seconds again a 2-hop connection is established between Node 0 and Node 1 which lasts very short for about 9 seconds when the Node 2 moves out of the range of the Node 1 and the connection is broken and the TCP window size again shrinks to zero. This event is depicted in Figure. 6.

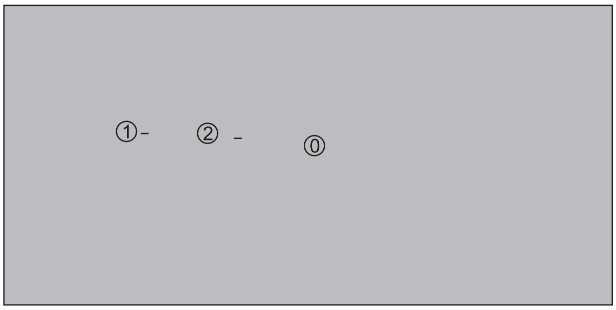


Figure. 6: A Short Span of Second Phase of Two Hop Connection Between the Nodes 0 and 1 At Around 235 Secs.

The whole variation in the TCP window size is depicted in Figure. 7 using the following command of the simulator:

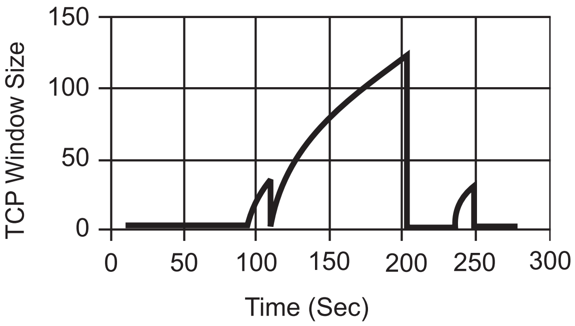


Figure. 7: TCP Window Size Evaluation for DSDV protocol.

$ns at 10.1 “plotWindow $tcp $windowVsTime”

Where the resolution of the window size plot is already kept at 0.01.

In order to see how many of the send packets have been received, enqued, dequeued or dropped we have to look at the ASCII version of the events log file that has been generated during the whole process of connection make and break. Network Simulator-2.35 not only gives us a pictorial version of the events display in the form of Network AniMator (NAM) but also gives us the ASCII version of the traces of all the events occurring into the system. This ASCII version of the events information is stored in the form of a trace file. The trace file entries for the simulated model were of the following form:

r 162.804394634 _1_ AGT — 9715 tcp 1060 [13a 1 0 800] ——— [0:0 1:0 32 1] [4834 0] 1 0

All the trace entries were consisting of a number of fields, each having particular information regarding the traced events.

After the traces of a file are generated depicting different events occurring during execution of the .tcl script, we have used GNU AWK[17] (GAWK), text-manipulation and pattern-scanning language, in order to extract the information of our use from these traces. GNU AWK, often called data-driven language is a powerful Open-source text-processing and pattern-matching language used for scanning particular text patterns and isolating them from rest of the text in the provided data[18]. In order to evaluate how many tcp packet have been delivered by the DSDV protocol, we have used AWK script at the BASH terminal. It shows that the number of TCP packets delivered by DSDV protocol are 8489 in the stipulated simulation time of 280 secs.

The same parameters when applied to the reactive protocol DSR, produces the window size as shown in Figure 8.

Figure 8 reveals that the communication between the nodes is started at around 80 secs which is a bit earlier compared to the case when we used DSDV protocol with the same parameters. A packet drop indicated by a sudden fall in the size of tcp window at around 110 secs is can be seen when the communication is shifted from 2-hop to 1-hop due to the proximity of the two nodes during the random movement and then shows a continuous trend of rise till there is a slight change in the trend of the graph depicted by a linearity region starting at around 204 secs which is due to the fact that the communication is shifted again from 2-hop to 1-hop. At around 248 seconds when the two nodes, during there random course of movement goes out of range, there window size is dropped down to zero as communication is not possible between the two nodes. While implementing the model for DSR protocol, its necessary to use the Carnegie Mellon University (CMU's) wireless extension to NS-2 (incorporated in the release NS-2.1b9a). This extension is necessary to be implemented at the Interface Queue type for DSR. CMUPriQueue classifies the packets into four queues namely video, audio, control packet and rest of traffic. Remaining parameters remain the same for DSR.

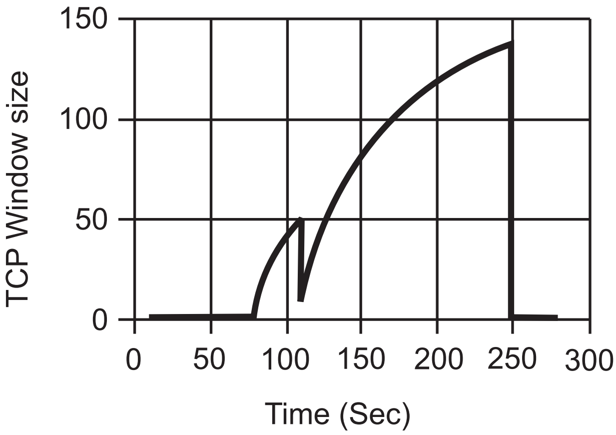


Figure 8: TCP Window Size Evaluation for DSR protocol.

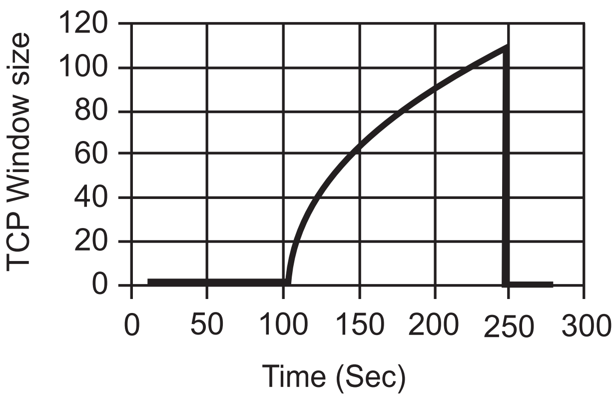


Figure 9: TCP Window Size Evaluation for AODV protocol.

Using the same movement scenario as described for DSDV, the number of received tcp packets for DSR is 10470. Proactive protocol DSDV utilizes a considerable amount of packets inefficiently by constantly updating its routing paths even if these are not used thus it results in dropage of packets.

When we evaluate AODV protocol using the same parameters, we see that the window size evaluation does not encounter any break during the whole course of movement scenario as indicated by Figure 9. 2-Hop path is used for the whole communication session starting from 105 seconds, late than both the DSDV and DSR protocol. The tcp window size continues to rise till 247 secs when the two nodes get out of each other's range and the tcp window size drops down to zero. The continuous increase in window size is due to the fact that there is no break in the communication session during the whole course of movement scenario and a 2-hop connection is established throughout the communication session between the two nodes due to the which the Reno/Tahoe algorithm implemented at the tcp level keeps on increasing the window size in the integer units of MSS (Maximum Size Segment) till a loss event occurs which is detected by a time-out event or reception of three duplicate acks after which the congestion window size is dropped down to least value. The number of tcp packets delivered during the communication session for AODV protocol is 5995. This is the least of the three as depicted in Figure 10.

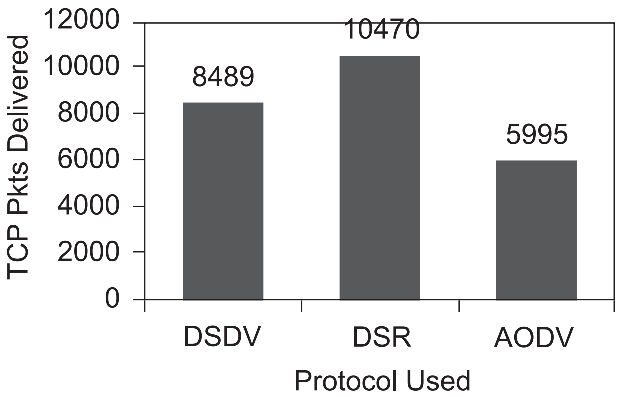


Figure 10: Number of TCP packets delivered to the destination for each protocol.

Reason for the least reception of the tcp packets is the round-trip time which is elapsed while packets are transmitted through 2-hops. Though the window size keeps on increasing as there is no break in the communication scenario from 1-hop to 2-hops or vice-versa but due to the time taken by the acks to traverse 2 hops, limits the number of transmitted packets because it is only after reception of the acks that the implemented algorithm increases the congestion window size in the integer units of MSS. And it is the basic reason that the AODV transfers less data then DSR or DSDV under the same simulation parameters and movement scenario. Thus we infer that a routing path with less number of hops results in a better TCP performance due to the less round-trip time elapsed by the receiving Acks and transmitted data.

CONCLUSION

A performance evaluation of the three most widely used MANET protocols is presented based on there TCP window size evaluation. The results show that using the same simulation parameters the DSR prtocols outperforms the distance vector based routing protcols in terms of the number of tcp packets successfuly transmitted to the receiving node. As far as the duration of the communicaiton session is concerned, again DSR serves the longest communication session between the nodes with a few drops in the tcp window sizes when compared with the competitors.

ACKNOWLEDGMENT

Authors are thankful to Higher Education Commission Pakistan who supported this work under the Program “HEC Indigenous Fellowships Scheme for PhD Scholars, Batch-IV.” Authors’ld also like to show their gratitute to the respectable reviewers who gave their expert opinion to improve the quality of the work.

REFERENCES

1 *Caihong K, Nenghai Yu and Chen Yuzhong C. 2007 “An enhancement scheme of TCP protocol in mobile ad hoc networks: MME-TCP1”. Springerlink Jour. Of Elec.(China), 24(1): 1-9.*

2 *Kant L. et. al., 2008 “Network Science Based Approaches to Design and Analyze MANETs for Military Applications”, IEEE Comm. Magazine, 46(11), 55-61.*

3 *Barrere L., Chaumette S. and J. Turbert, “A tactical active information sharing system for military manets”, in proc. of MILCOM 2006 (Washington), October 23-25, 2006.*

4 *Misra R. and Manda1 2005, “Performance comparison of AODV/DSR on-demand routing protocols for ad hoc networks in constrained situation”, in the proc. Of IEEE Int. Conf. on Personal Wireless Comm., pp:86-89, Jan 23-25.*

5 *Park S. and Yoo S. Seong-Moo Yoo, 2010 “Routing table maintenance in mobile ad hoc networks”, in the proc. of IEEE 12th Int. Conf. on Advanced Comm. Tech. (ICACT), pp: 1321-1325.*

6 *Qadri N.B. and G.A. Shah, 2010 “Performance Evaluation of Ad-hoc Routing Protocols in Underwater Acoustic Sensor Networks”, in the proc. Of IEEE 19th annual Wireless and Optical Comm. Conf., pp: 1-6.*

7 *Perkins C. and Das S. 2003 “Ad hoc On-Demand Distance Vector (AODV) Routing,” IETF RFC 3561.*

8 *Usop N.S.M et al., 2009 "Performance Evaluation of AODV, DSDV & DSR Routing Protocol in Grid Environment", IJCSNS Int. Journal of Comp. Sc. And Network Security, vol. 9, No. 7, pp: 261-268, July 2009.*

9 *Gupta A.K., Sadawarti H. and Verma A.K., 2010. “Performance analysis of AODV, DSR & TORA Routing Protocols,” IACSIT Int. Journal of Engg. & Tech..,, vol. 2, No. 2, pp: 226-231.*

10 *Kim K.W., Lorenz P., Lee M.M.O., 2005 "A New Tuning Maximum Congestion Window for Improving TCP Performance in MANET," in Proc. 2005 Sys. Comm., ICW’05, pp. 73-78.*

11 *Mbarushimana C. and Shahrabi A., 2007 "Comparative Study of Reactive and Proactive Routing Protocols Performance in Mobile Ad Hoc Networks," in proc. Of IEEE 21st Int. Conf. on*

Adv. Info. Networking and Application Workshop (AINAW, 07), pp:679-684.

12 *Holland G., Vaidya N., 1999. "Analysis of TCP Performance over Mobile Ad Hoc Networks," in Proc. 5th Int. Conf. Mob. Comp. and Networking, Mobicom'99, pp. 219-230.*

13 *Papanastasiou S., Ould-Khaoua M., Mackenzie L., 2005 "On the Evaluation of TCP in MANETS," in* Proc. Int. Workshop Wireless Ad Hoc Networks.

14 *Chen K., Xue Y., Shah S.H. and Nahrstedt K. 2004. "Understanding bandwidth-delay product in mobile ad hoc networks,"* Elsevier Jour. Comp. Comm., *vol. 27, no. 10, pp. 923-924.*

15 *Das S.R., Castaiieda R., Yan J. & Sengupta R., 1998. "Comparative Performance Evaluation of Routing Protocols for Mobile, Ad hoc Networks,"* in the proc. Of IEEE 7th Int. Conf. on Comp. Comm. And Networking, *pp:153-161.*

16 *Alaettinoglu C., Shankar A.U., Zieger K.D. and Matta I., 1994. "Design and implementation of MaRS: A routing testbed,"* Jour. of Internetworking: Research and Experience, *vol. 5, No. 1, pp: 17-41.*

17 *Stutz M., 2006 "Get started with GAWK: AWK language fundamentals,"* http://www.ibm.com/developerworks/aix/tutorials/au-gawk/.