International Journal of Digital Content Technology and its Applications
Volume 3, Number 1, March 2009

TCP Performance Analysis in Mobile Ad Hoc Networks with Different Routing Protocols and Varying Payload

Noor Mast[1,2], Muhammad Nisar Afridi, and KoK-Keong Loo[1]

[1]School of Engineering and Design, Brunel University, UK
[2]Kohat University of Science and Technology (KUST), N.W.F.P, Pakistan

doi: 10.4156/jdcta.vol3.issue1.noor

Abstract

Mobile Ad hoc network routing protocols are classified into several categories such as proactive and reactive routing protocols. The performances of these categories are evaluated in different situations. This work presents TCP performance evaluation in three different scenarios with variable amount of payload and number of nodes. It was found that different amount of payload have different effects on the TCP performance. The traffic consists of three different packet sizes i.e. 512, 1000, 1500. Three different routing protocols (AODV, DSDV, and DSR) are evaluated with three different TCP variants (Sack, New Reno and Tahoe). The performances parameters on the basis of which routing protocols are graded are mainly throughput; others are congestion window and end to end delay. Our results indicate that DSR performs well in a variety of conditions.

1. Introduction

A Mobile Ad-hoc Network (MANET) is a temporary wireless network composed of mobile nodes without any permanent infrastructure. Each node not only operates as an end system, it also acts as a router to forward packets on behalf of other nodes [1]. One of the best feature of MANET is its flexibility and can configure itself in the fly and thus very suitable for the emergency situation.

The IEEE 802.11 MAC protocol[2] is a best choice for providing Ad-hoc network facilities. Mobile ad hoc network can be a standalone network or it is also possible to connect it to the infrastructure network. Thus it provide the facility to connect to the internet from anywhere. According to [1] typical applications of MANET are

(1) Military use
(2) Search and rescue.
(3) Vehicle-to-vehicle communication in intelligent transportation.
(4) Temporary networks in meeting rooms, airports, etc.
(5) Personal Area Networks connecting cell phones, laptops, smart watches, and other wearable computers

2. Background

In this section we are presenting a brief overview of the three routing protocols DSDV, DSR and AODV as well as the different variants of TCP to be analyzed.

2.1 Routing Algorithms

2.1.1 Destination Sequenced Distance Vector (DSDV)

DSDV [3] is a hop-by-hop distance vector routing protocol that in each node has a routing table that for all reachable destinations stores the next-hop and number of hops for that destination.

2.1.2 Ad-hoc On Demand Distance vector (AODV)

The Ad hoc On-Demand Distance Vector (AODV) algorithm enables dynamic, self-starting, multi-hop routing between participating mobile nodes wishing to establish and maintain an Ad hoc network [4].

2.1.3 Dynamic Source Routing (DSR)

Dynamic Source Routing (DSR) [5] also belongs to the class of reactive protocols and allows nodes to dynamically discover a route across multiple network hops to any destination.

2.2 Transmission Control Protocol - TCP

TCP has changed slowly and progressively since its initial introduction. The following are the main changes which were introduced in the classical TCP.

2.2.1 Tahoe TCP

In 1988, Jacobson introduced the congestion control algorithm [6] in TCP and it is called TCP-Tahoe. It was assumed by Jacobson that losses due to packet corruption are much less probable than losses due to buffer overflows on the network. Therefore, on a loss, the sender should lower its share of the bandwidth. Tahoe TCP includes Slow-Start, Congestion Avoidance and Fast Retransmit. Tahoe TCP detects packet loss by either expiring RTO or receiving 3 duplicate ACKs. In case of receiving 3 duplicate ACKs Tahoe TCP retransmits the lost packet without having to wait for the RTO to expire. As RTO is relatively quite enough to transmit a packet, this process is called Fast Retransmit. Upon receiving a congestion signal Tahoe TCP sets its threshold value to half of the value of the congestion window just before the lost is recorded and congestion window is set to 1 MSS and enters into slow start. This process is called congestion avoidance. Tahoe TCP does not deal well with multiple packet loss within a single window of data.

2.2.2 Reno TCP

TCP Reno [7] introduced a major improvement over Tahoe TCP by introducing Fast Recovery algorithm. Upon receiving duplicate ACKs, Reno TCP does not enter into Slow-Start but it dictates the sender to perform the congestion avoidance mechanism after fast retransmission.

2.2.3 New Reno TCP

NewReno [8] modifies the Fast Recovery algorithm in such a way to deal with multiple packet losses in a single congestion window.

2.2.4 S ACK TCP

One of the reason of TCP poor performance is that TCP's fast retransmission and fast recovery mechanism can handle one packet loss from one window of data. To overcome this limitation and handle multiple packet loss in the same window the selective acknowledgement option (SACK)[9] was introduced in standard TCP implementation.

3 Simulations

In this research the simulator that we have used is the Network Simulator 2 (NS-2) from Berkeley [10]. To simulate the mobile wireless radio environment we have used a mobility extension to NS that is developed by the CMU Monarch project at Carnegie Mellon University. NS-2 along with CMU extension provides support for all TCP implementation and routing protocols (AODV, DSDV and DSR) that we have used in this research paper. Also all of these protocols are used with their basic configurations and no changes have been made.

3.1 Problem Definition

In this paper we are interested in finding out that how back ground traffic affects the relative performance of Mobile Ad hoc routing protocols. Also to make ostensible the background traffic consists of three different packet sizes i.e. 512, 1000, 1500 bytes and three different scenarios of 3 , 5 and 10 nodes. Selected routing protocols (AODV, DSDV and DSR) are simulated with different TCP variants (New Reno, Sack and Tahoe). In each scenario the three selected routing protocols along with three TCP variants are simulated having three AM in a different back ground traffic. Routing protocols performance is evaluated on the bases of throughput, number of packets sent, number of acknowledgement received, round trip time etc. we have collected data about these parameters and concluded our results.

3.2 Simulation scenario

In this research, we have taken three scenarios of 3 nodes, 5 nodes and 10 nodes. In our first simulation scenario, we have taken three wireless nodes over the area of a size of 500m x 400m. A TCP connection is established between the node(0) and node(1) with having background traffic. The initial location of the node(0), node(1) and node(2) as shown in Figure: 1, are receptively (5, 5), (490, 285) and (150, 240), the z co-ordinate is assumed to be 0.

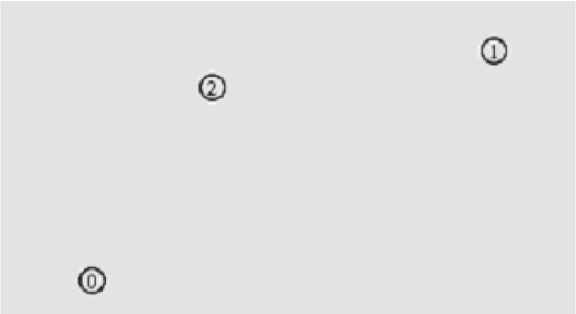


Figure 1. Three Nodes Initial Position.

In the second scenario, we have taken five wireless nodes over the area of a size of 500m x 400m. A TCP connection is established between the node(0) and node(1). The initial location of the node(0), node(1) and node(2) same as in first scenario shown in Figure: 2, are receptively (5, 5), (490, 285) and (150, 240), the z co-ordinate is assumed to be 0. The other two nodes (3 and 4) are positioned at co-ordinates (400,11), (10,120) respectfully. All the nodes produce back ground traffic.

Figure 2. Five Nodes Initial Position.

Finally our third scenario consists of 10 wireless nodes over the area of a size of 500m x 400m. A TCP connection is established between the node(0) and node(1).The initial position of the nodes in the scenario are shown in Figure: 3. As mentioned afore, all the nodes produce back ground traffic.

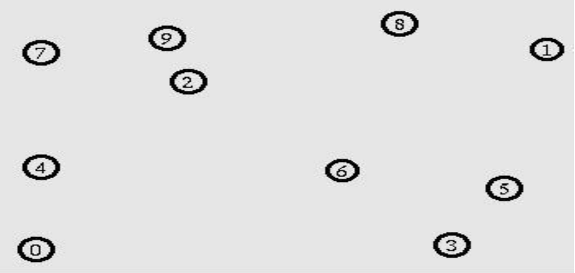


Figure 3. Ten Nodes Initial Position

3.3 Traffic pattern

The traffic pattern defines the way in which the mobile node moves in a scenario. In our analysis we have taken three scenarios of 3, 5 and 10 nodes. The traffic pattern for the three mobile nodes in the simulation scenario is defined in such away that they remain in each other's radio range for the maximum of simulation time, so that the TCP connection should be remained open. The traffic pattern is: at time 10, node(0) starts moving towards point (250, 250) at the speed of 3 m/sec, at time 15, node (1) starts moving towards point (48, 285) at a speed of 5 m/sec, at time 110, node(0)starts moving towards point (480, 300) at a speed of 5 m/sec and at time 110, node(1) starts moving towards point (10,150)at a speed of 5 m/sec. Node(2) is kept static in the simulation. It will just act as a relay between node(0) and node(1) to help in increasing the connection time and introduces an imaginary propagation and processing delay. The duration of the simulation is kept 155 seconds, as it is the maximum time to get the required results.

The traffic pattern for the five nodes simulation scenario is again defined in such away that they remain in each other's radio range for the maximum of simulation time, so that the TCP connection should be remained open. The traffic pattern is: at time 5, node(3) starts moving towards point (240,185) at the speed of 5 m/sec, at time 10, node (4) starts moving towards point (80,130) at a speed of 5 m/sec, and node(0) start moving towards points (250,250). At time 15 node(1) starts moving towards (45,285) at a speed of 5 m/sec. Similarly at time 50 node(4) moves to words co-ordinates (380,125). At time 80 node(3) moves towards points (280,290). Node(0) moves towards (480,300) and node(1) moves towards coordinates(10,150) at time 110 at speed of 5 m/sec. In order to get the desirable proportionality we have kept the duration of simulation constant (i.e. 155 sec). Finally the traffic pattern for the ten mobile nodes in the simulation scenario is defined in such away that they again remain in each other's radio range for the maximum of simulation time. The traffic pattern is: : at time 5, node(3) starts moving towards point (240,185) at the speed of 5 m/sec, at time 10, node (4) starts moving towards point (80,130) at a speed of 5 m/sec, and node(0) start moving towards points (250,250). At time 15 node(1) starts moving towards (45,285) at a speed of 5 m/sec. At time 25, node(7)starts moving towards point (4,125) at the speed of 5 m/sec, at time 30 node (5) starts moving towards point (340,125). At time 50 and 55, node(4) and node(7) starts moving towards point (380,125) and (4,125) respectfully at a speed of 5 m/sec. At time 60 and 70 node(8) and node(9) moves towards (165,105) and (310,95) respectfully at same speed 5 m/sec. Similarly at time 80, 85 and 90 node(3), node(5) and node(6) moves towards co-ordinates (280,290), (140,6) and (8,4) at a speed of 5 m/sec. Node(0) and node(1) starts moving towards point (480,300) and (10,150) at time 110 and at a speed of 5 m/sec. Again the duration of the simulation is kept 155 seconds.

4 Connection Pattern

The connection pattern defines that who is connected to whom in the simulation scenario. In simulation scenario a TCP connection is established between node(0) and node(1). The background traffic is discussed later on. Node(0) acts as TCP source and node(1) acts as a TCP sink. TCP source sends TCP packets and TCP sink sends acknowledgements back to the source. The TCP packet size is set to its original length (1000Kb). Node(0) sends FTP packets. At time 12.0 FTP transmission starts, but when node(0) and node(1) comes in each other radio range or in the radio range of other nodes so that they can established a TCP connection, a TCP connection will be created and data transmission will take place.

4.1 Background Traffic

Most of the research work has ignored the real world problems and therefore they can not model well the real world environment. This simulation is designed with real world interference in mind caused by background traffic and other factors. As we have mentioned earlier that our simulation is consist of three scenarios of 3, 5 and 10 nodes.

The background traffic consists of three different packet sizes and hence different amount of background traffic. In 3 nodes scenario, when the packet size is 512 bytes then background traffic comes out 67584 bytes. When the packet size is 1000 bytes then background traffic equals to 132000 bytes. When the packet size is 1500 bytes then background traffic equals to 198000 bytes. Similarly in 5 nodes scenario for packet size 512, 1000 and 1500 the background traffic are 92672, 21600 and 324000 respectfully. And for 10 nodes scenario when the packet sizes are 512, 1000 and 1500, the background traffic is 205312, 376000 and 564000 respectfully. These three background traffic are taken in each of 3, 5 and 10 nodes scenarios. The three nodes scenario is shown in figure. The dotted lines indicate the connection between nodes.

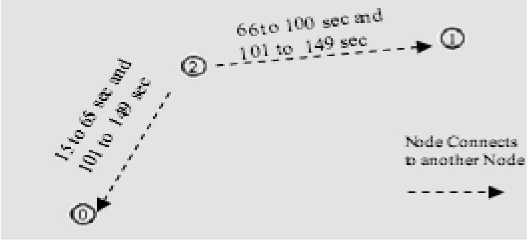


Figure 4 Three Nodes UDP Connection Setup

At 15 second just after the FTP traffic starts at 12 second node(2) establish a UDP connection with node(0) and starts communicating up to 65 second at a constant bit rate. After this at 66 second node(2) initiates a UDP connection with node(1) up to 100 second. Similarly a third UDP connection is established between node(0) and node(1) from 101 to 149 second. The background traffic is designed in such a way to provide maximum interference to the communication nodes. The 5 nodes scenario is shown in figure 5.

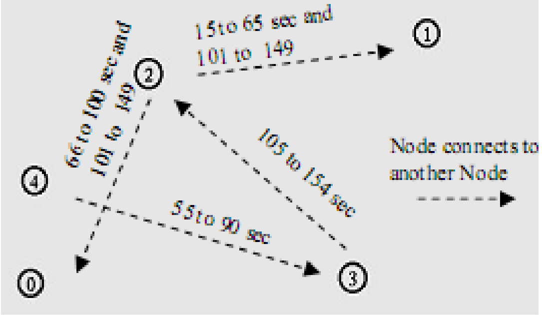


Figure 5. Five Nodes UDP Connection setup

Now again at 15 second node(2) establish a UDP connection with node(0) and starts communicating up to 65 second at a constant bit rate. After this at 55 second node(4) initiates a UDP connection with node(3) up to 100 second. At 66 second node (2) established UDP connection to node (0). At 101 node(2) again initiate UDP connection to node(0)and node(1) till 149 second. Similarly at node (3) established UDP connection to node (2) from 105 to 154 second.

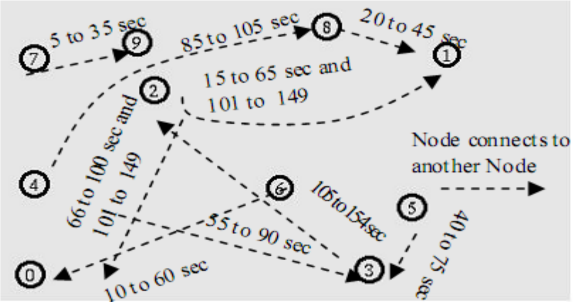


Figure 6. Ten Nodes UDP Connection Setup

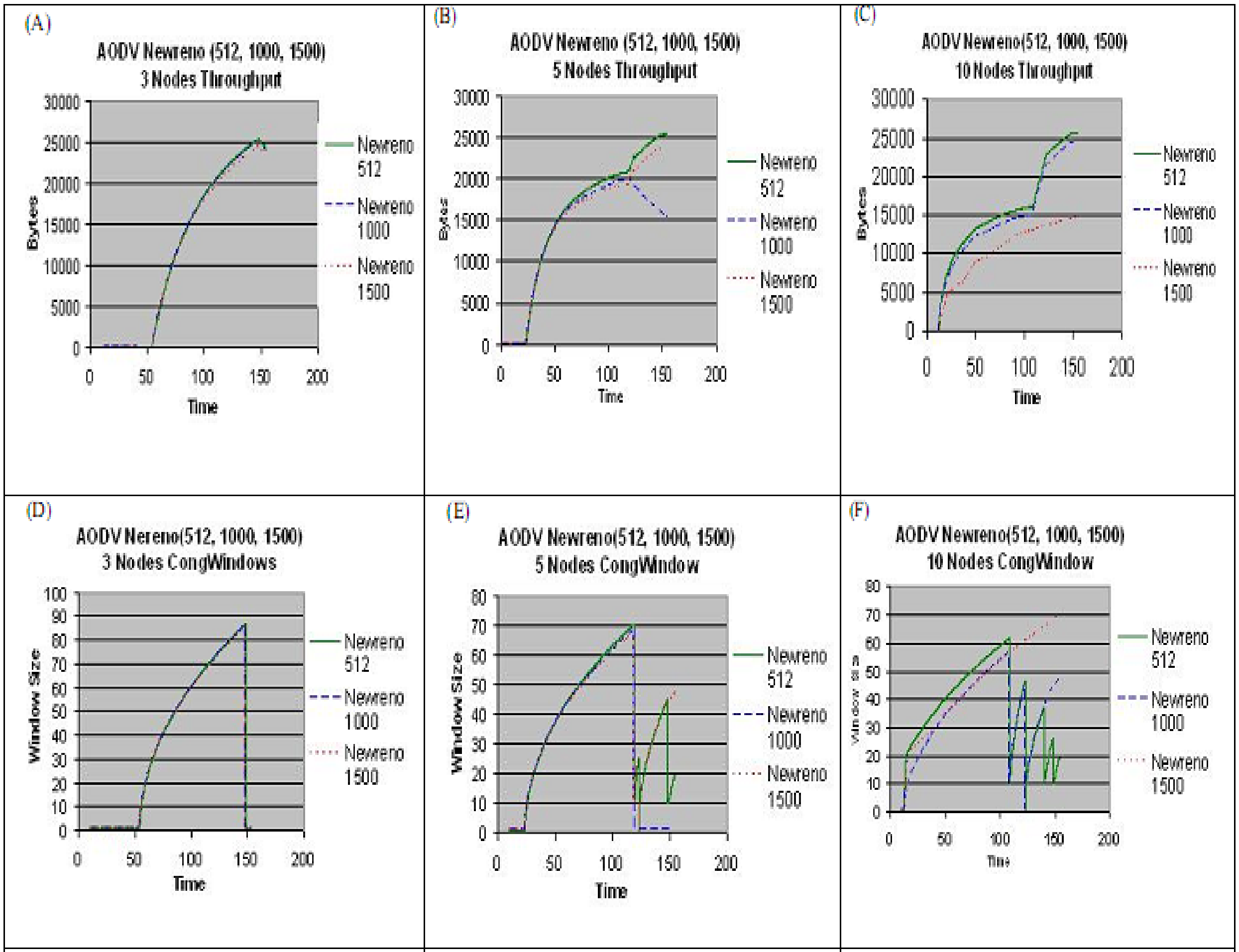
This time at 5 seconds node (7) establish a UDP Connection with node (9) and starts communicating up to 35 second at a constant bit rate. Then at 10 second node(6) connects to node(0) till 60 seconds. After this at 15 second node(2) initiates a UDP connection with node(1) up to 65 second and then again make connection at 101 seconds till 149 seconds. At 20 second node(8) established UDP connection to node(1) up to 45 seconds. At 40 node(5) initiate UDP connection to node(8) up to 75 seconds. At 55 seconds node(4) connects to node(3) up till 80 second and node(2) connects to node(1) at 66 up to 100 seconds. Similarly node(4) initiate UPD connection with node(8) from 85 till 105 second and node(3) established UDP connection with node(2) at 105 seconds up to 154 seconds.

5 Results

This section is reserved to discuss the simulation results..

5.1 TCP Algorithms with AODV

When AODV is simulated with different TCP algorithms, the results obtained are show in Figure 7.



(A). AODV New Reno Throughput (512, 1000 and 1500 packet size) 3 nodes scenario.

(B). AODV New Reno Throughput (512, 1000 and 1500 packet size) 5 nodes scenario.

). AODV New Reno Throughput (512, 1000 and 1500 packet size) 10 nodes scenario.

(D). AODV New Reno CongWindow (512, 1000 and 1500 packet size) 3 nodes scenario.

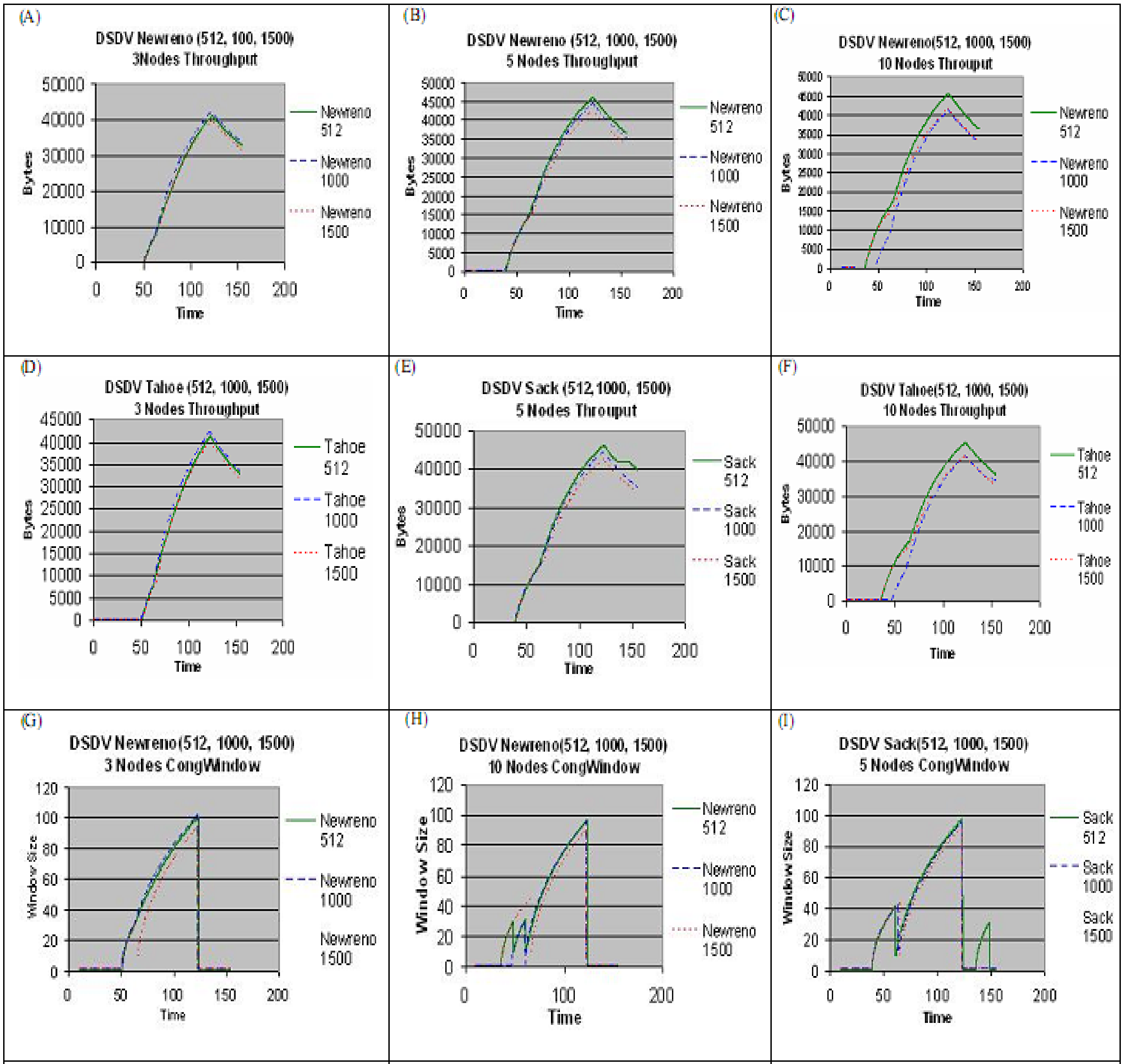
(E). AODV New Reno CongWindow (512, 1000 and 1500 packet size) 3 nodes scenario.

F). AODV New Reno CongWindow (512, 1000 and 1500 packet size) 3 nodes scenario.

Figure 7.

5.2 TCP algorithms with DSDV

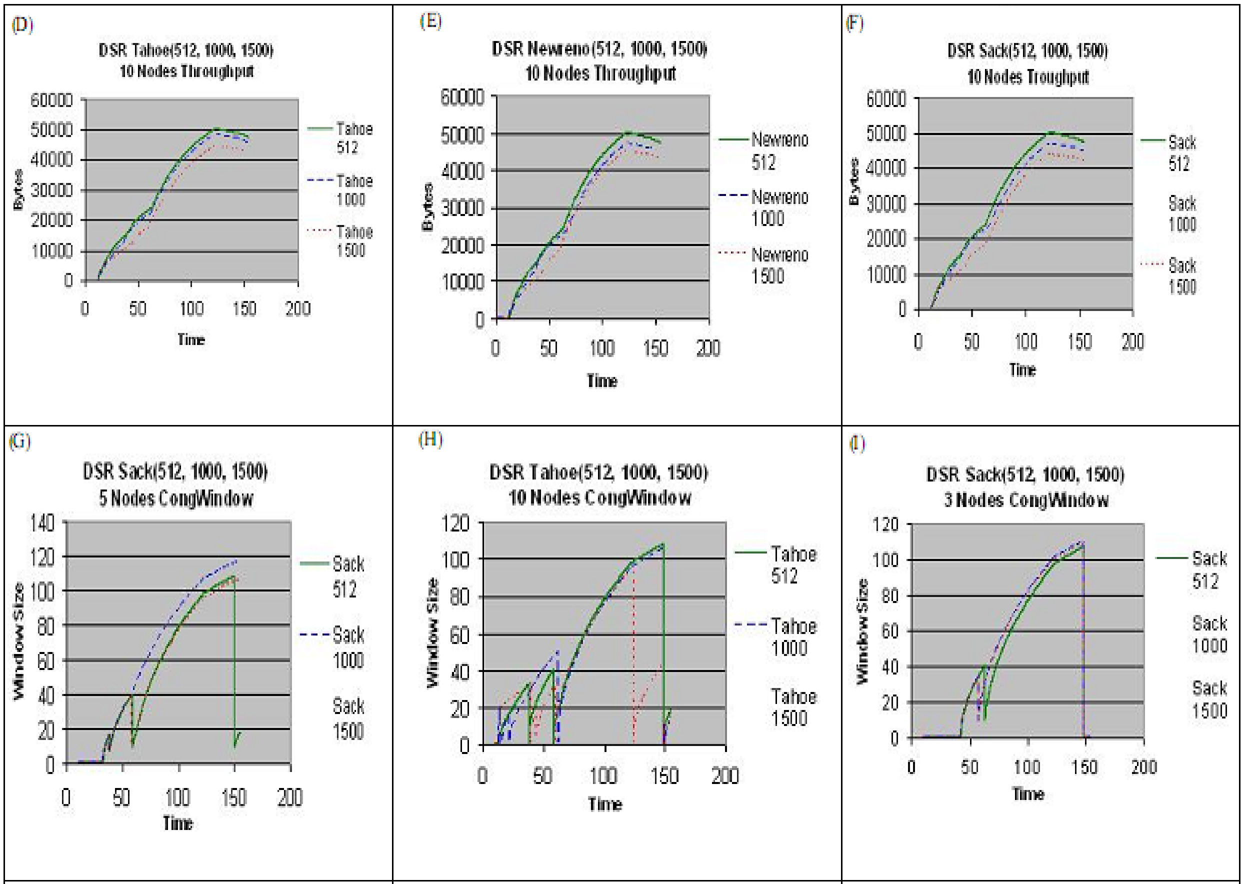
When DSDV was simulated with the three algorithms of TCP the following results were obtained.



(A) DSDV New Reno Throughput (512, 1000 and 1500 packet size) 3 nodes scenario	**(B)** DSDV New Reno Throughput (512, 1000 and 1500 packet size) 5 nodes scenario	**(C)** DSDV New Reno Throughput (512, 1000 and 1500 packet size) 10 nodes scenario
(D) DSDV Tahoe Throughput (512, 1000 and 1500 packet size) 3 nodes scenario	**(E)** DSDV Sack Throughput (512, 1000 and 1500 packet size) 5 nodes scenario	**(F)** DSDV Tahoe Throughput (512, 1000 and 1500 packet size) 10 nodes scenario
(G) DSDV New Reno CongWindow (512, 1000 and 1500 packet size) 3 nodes scenario	**(H)** DSDV New Reno CongWindow (512, 1000 and 1500 packet size) 10 nodes scenario	**(I)** DSDV Sack CongWindow (512, 1000 and 1500 packet size) 5 nodes scenario

Figure 8.

5.3 TCP Algorithms with DSR



(A) DSR Sack Throughput (512, 1000 and 1500 packet size) 3 nodes scenario.
(D) DSR Tahoe Throughput (512, 1000 and 1500 packet size) 10 nodes scenario.
(G) DSR Sack CongWindow (512, 1000 and 1500 packet size) 5 nodes scenario

(B) DSR Sack Throughput (512, 1000 and 1500 packet size) 5 nodes scenario.
(E) DSR New Reno Throughput (512, 1000 and 1500 packet size) 10 nodes scenario.
(H) DSR Tahoe CongWindow (512, 1000 and 1500 packet size) 10 nodes scenario

C) DSR New Reno Throughput (512, 1000 and 1500 packet size) 5 nodes scenario.
(F) DSR Sack Throughput (512, 1000 and 1500 packet size) 10 nodes scenario.
(I) DSR Sack CongWindow (512, 1000 and 1500 packet size) 3 nodes scenario

Figure 9.

5.4 Results Comparison

(A). AODV

Analysis of AODV routing protocol with three TCP algorithms (New Reno, Sack and Tahoe) having three different scenarios of 3, 5 and 10 nodes and three different background traffic in each scenario. The table summarizes the result for AODV by varying the TCP algorithms and packet size keeping the number of nodes constant.

Table 1. AODV Analysis in 3 Nodes Scenario

S. No	Routing Algorithm	TCP Standards	Background Traffic	No. of Nodes	Throughput
1	AODV	New Reno	512	3	24267.27
2	AODV	Sack	512	3	24267.27
3	AODV	Tahoe	512	3	24267.27
4	AODV	New Reno	1000	3	23905.75
5	AODV	Sack	1000	3	23905.75
6	AODV	Tahoe	1000	3	23905.75
7	AODV	New Reno	1500	3	23324.73
8	AODV	Sack	1500	3	23324.73
9	AODV	Tahoe	1500	3	23324.73

After running the simulation it was found that AODV produces the same throughput with the investigated TCP flavors (New Reno, Sack and Tahoe) with same back ground traffic. The background traffic was varied in the simulation by varying the packet size. I have divided the result in three different sections.

1. 1st case

a). Number of nodes 3
b). Packet Size 512

Now having set these parameters when AODV was simulated with the different TCP algorithms (New Reno, Sack and Tahoe) it was found that AODV produces the same throughput with all the three flavors of TCP. In this case the throughput is 24267.27 bytes.

Table 2. AODV Analysis in 5 Nodes Scenario

S. No	Routing Algorithm	TCP Standards	Background Traffic	No. of Nodes	Throughput
1	AODV	New Reno	512	5	25397.03
2	AODV	Sack	512	5	24699.81
3	AODV	Tahoe	512	5	25132.34
4	AODV	New Reno	1000	5	15326.02
5	AODV	Sack	1000	5	23808.91
6	AODV	Tahoe	1000	5	15326.02
7	AODV	New Reno	1500	5	23763.72
8	AODV	Sack	1500	5	23105.23
9	AODV	Tahoe	1500	5	23660.43

3. 2nd case

a). Number of nodes 3
b). Packet Size 1000

Now again AODV was found consistent with the mentioned flavours of TCP. AODV produced same throughput with all flavours but this time smaller than the first case. In this case the throughput is 23905.75 bytes.

3. 3rd case

a). Number of nodes 3
b). Packet Size 1500

Once again AODV was found consistent with the said TCP algorithms. AODV produced same throughput with all flavors but this time smaller than the first case. In this case the throughput is 23324.73 bytes. The following table summarizes the result for AODV by varying the TCP algorithms and packet size keeping the number of nodes constant (i.e. 5).

4. 4th case

a). Number of nodes 5
b). Packet Size 512

Now having set these parameters when AODV was simulated with the different TCP algorithms (New Reno, Sack and Tahoe) it was found that AODV produces the different throughput with all the three flavors of TCP. In this case AODV produces maximum throughput New Reno which is 25397.03 bytes.

5. 5th case

a). Number of nodes 5
b). Packet Size 1000

With having 216000 bytes background traffic and 5 nodes, AODV again produces different throughput with the mentioned TCP algorithms (New Reno, Sack, and Tahoe). Now this time AODV produces maximum throughput with Sack, that is 23808.91 bytes. The throughput is less than the throughput when the packet size was 512 bytes.

6. 6th case

a). Number of nodes 5
b). Packet Size 1500

Similarly having changing the background traffic to 324000, AODV again produces maximum throughput with New Reno and is 23763.72 bytes, which is also less than the throughput when the packet size was 512 bytes. The following table summarizes the result for AODV by varying the TCP algorithms and packet size keeping the number of nodes constant to 10.

Table 3. AODV Analysis in 10 Nodes

S. No	Routing Algorithm	TCP Standards	Background Traffic	No. of Nodes	Throughput
1	AODV	New Reno	512	10	25694
2	AODV	Sack	512	10	22898.64
3	AODV	Tahoe	512	10	25655.26
4	AODV	New Reno	1000	10	24467.4
5	AODV	Sack	1000	10	24460.94
6	AODV	Tahoe	1000	10	24370.56
7	AODV	New Reno	1500	10	14732.09
8	AODV	Sack	1500	10	14732.09
9	AODV	Tahoe	1500	10	14732.09

7. 7th case

a). Number of nodes 10
b). Packet Size 512

Finally the number of nodes is set to 10 and AODV was simulated with the mentioned 3 TCP algorithm (New Reno, Sack and Tahoe) with varying background traffic. In this case the background traffic is 205312 bytes and AODV produces maximum throughput with New Reno. The throughput is 25694 bytes.

8. 8th case

a). Number of nodes 10
b). Packet Size 1000

With 376000 bytes background traffic AODV produces maximum throughput again with New Reno. The throughput is 24467.4 bytes.

9. 9th case

a). Number of nodes 10
b). Packet Size 1500

In last case of AODV when the background traffic is set to 564000 bytes, AODV produces maximum throughput with Sack that is 14732.09 bytes.

Table 4. DSDV Analysis in 3 Nodes

S. No	Routing Algorithm	TCP Standards	Background Traffic	No. of Nodes	Throughput
1	DSDV	New Reno	512	3	32743.71
2	DSDV	Sack	512	3	32905.1
3	DSDV	Tahoe	512	3	32743.71
4	DSDV	New Reno	1000	3	33479.66
5	DSDV	Sack	1000	3	33408.65
6	DSDV	Tahoe	1000	3	33479.66
7	DSDV	New Reno	1500	3	31594.58
8	DSDV	Sack	1500	3	31710.78
9	DSDV	Tahoe	1500	3	31678.5

(B) DSDV

Now we will take the 2nd routing protocol DSDV and again three TCP algorithms (New Reno, Sack and Tahoe) to see the throughput by varying the number of nodes to 3, 5, 10 and the background traffic. Again we have divided the DSDV simulation in 9 cases for better explanation. Now this table summarizes the result of DSDV by varying the TCP algorithms and packet size keeping the number of nodes constant to 3.

1. 1st case

a). Number of nodes 3
b). Packet Size 512

When DSDV was simulated with mentioned TCP algorithms (New Reno, Sack and Tahoe) and the number of nodes is kept to 3 and 67584 bytes background traffic, DSDV produces same throughput with New Reno and Tahoe but Different with Sack. The maximum throughput is with New Reno and Tahoe. The Throughput is 32743.71 bytes.

2. 2nd case

a). Number of nodes 3
b). Packet Size 1000

In this case DSDV produces maximum throughput with again New Reno and Tahoe. The throughput is 33479.66 bytes.

3. 3rd case

a). Number of nodes 3
b). Packet Size 1500

Similarly in this case DSDV produces maximum throughput with Sack and is 31710.78 bytes. The table summarizes the result of DSDV by varying the TCP algorithms and packet size keeping the number of nodes constant (i.e. 5).

Table 5. DSDV Analysis in 5 Nodes

S. No	Routing Algorithm	TCP Standards	Background Traffic	No. of Nodes	Throughput
1	DSDV	New Reno	512	5	36578.44
2	DSDV	Sack	512	5	39761.14
3	DSDV	Tahoe	512	5	36559.07
4	DSDV	New Reno	1000	5	35209.81
5	DSDV	Sack	1000	5	35196.9
6	DSDV	Tahoe	1000	5	35183.99
7	DSDV	New Reno	1500	5	33589.41
8	DSDV	Sack	1500	5	33673.34
9	DSDV	Tahoe	1500	5	33628.15

4. 4th case

a). Number of nodes 5
b). Packet Size 512

Now once again the numbers of nodes are increased to 5 and the background traffic is varied. In this case the background traffic is 92672 bytes. DSDV produces different result with the mentioned TCP algorithms. The maximum throughput comes with Sack which is 39761.14 bytes.

5. 5th case

a). Number of nodes 5
b). Packet Size 1000

In this case the background traffic is increased keeping the number of nodes to 5. DSDV produces the maximum throughput with New Reno. The throughput is 35209.81 bytes.

6. 6th case

a). Number of nodes 5
b). Packet Size 1500

Again the number of nodes is kept 5 but the background traffic is taken 324000 bytes. DSDV produces maximum through with Sack which is 33673.34 bytes. The following table summarizes the result of DSDV by varying the TCP algorithms and packet size keeping the number of nodes constant this time to 10.

Table 6. DSDV Analysis in 10 Nodes

S. No	Routing Algorithm	TCP Standards	Background Traffic	No. of Nodes	Throughput
1	DSDV	New Reno	512	10	36281.47
2	DSDV	Sack	512	10	36184.64
3	DSDV	Tahoe	512	10	36178.18
4	DSDV	New Reno	1000	10	32730.79
5	DSDV	Sack	1000	10	32601.68
6	DSDV	Tahoe	1000	10	34125.24
7	DSDV	New Reno	1500	10	33021.3
8	DSDV	Sack	1500	10	32588.77
9	DSDV	Tahoe	1500	10	32990.96

7. 7th case

a). Number of nodes 10
b). Packet Size 512

Once again the numbers of nodes are increased to 10 and the background traffic is varied. DSDV produced different through with the three simulated TCP algorithms (New Reno, Sack and Tahoe).In this case DSDV produces maximum through with New Reno which is 36281.47 bytes.

8. 8th case

a). Number of nodes 10
b). Packet Size 1000

Similarly in this case DSDV produces maximum through with Tahoe which is 34125.24 bytes.

9. 9th case

a). Number of nodes 10
b). Packet Size 1500

Now in case of 564000 bytes background traffic and 10 nodes, DSDV produces maximum throughput with once again with New Reno which is 33021.3 bytes. Analysis of DSR routing protocol with three TCP algorithms (New Reno, Sack, Tahoe) having three different scenarios of 3, 5, 10 nodes and three different background traffic in each scenario.

Table 7. DSR Analysis in 3 Nodes Scenario

S. No	Routing Algorithm	TCP Standards	Background Traffic	No. of Nodes	Throughput
1	DSR	New Reno	512	3	42859.91
2	DSR	Sack	512	3	42892.19
3	DSR	Tahoe	512	3	42840.54
4	DSR	New Reno	1000	3	42969.66
5	DSR	Sack	1000	3	42427.37
6	DSR	Tahoe	1000	3	42943.83
7	DSR	New Reno	1500	3	41291.16
8	DSR	Sack	1500	3	43288.86
9	DSR	Tahoe	1500	3	41291.16

(C) DSR

Now this table summarizes the result of our last simulated routing protocol DSR by varying the TCP algorithms and packet size keeping the number of nodes constant (i.e. 3).

1. 1st case

a). Number of nodes 3
b). Packet Size 512

In this 1st case, DSR was simulated with the three mentioned TCP algorithms. The maximum throughput comes with Sack which is 42892.19 bytes.

2. 2nd case

a). Number of nodes 3
b). Packet Size 1000

Similarly in this case DSR gives maximum throughput with New Reno. The throughput is 42969.66 bytes.

Table 8. DSR Analysis in 5 Nodes

S. No	Routing Algorithm	TCP Standards	Background Traffic	No. of Nodes	Throughput
1	DSR	New Reno	512	5	45493.87
2	DSR	Sack	512	5	45274.37
3	DSR	Tahoe	512	5	45455.13
4	DSR	New Reno	1000	5	44086.51
5	DSR	Sack	1000	5	45500.32
6	DSR	Tahoe	1000	5	44563.04
7	DSR	New Reno	1500	5	42763.07
8	DSR	Sack	1500	5	42640.41
9	DSR	Tahoe	1500	5	42253.07

3. 3rd case

a). Number of nodes 3
b). Packet Size 1500

This time with 198000 bytes background traffic DSR produces maximum throughput with Sack which is 43288.86 bytes. The table summarizes the result of DSR routing protocol by varying the TCP algorithms and packet size keeping the number of nodes constant to 5.

4. 4th case

a). Number of nodes 5
b). Packet Size 512

Similarly the numbers of nodes are taken 5 and the background traffic is varied. The DSR routing protocol is simulated with same mentioned three TCP algorithms (New Reno, Sack and Tahoe) and the DSR produces different throughput with the three mentioned TCP algorithms. In this case DSR gives maximum throughput with New Reno which is 45493.87 bytes.

5. 5th case

a). Number of nodes 5
b). Packet Size 1000

In this case DSR gives maximum throughput with Sack. The throughput is 45500.32 bytes.

6. 6th case

a). Number of nodes 5
b). Packet Size 1500

In case of 324000 bytes background traffic the DSR produces maximum throughput with New Reno. The throughput is 42763.07 bytes. The following table summarizes the result of DSR routing protocol by varying the TCP algorithms and packet size keeping the number of nodes constant to 10.

Table 9. DSR Analysis in 10 Nodes

S. No	Routing Algorithm	TCP Standards	Background Traffic	No. of Nodes	Throughput
1	DSR	New Reno	512	10	47546.8
2	DSR	Sack	512	10	47579.08
3	DSR	Tahoe	512	10	47585.54
4	DSR	New Reno	1000	10	44848.29
5	DSR	Sack	1000	10	44816.01
6	DSR	Tahoe	1000	10	45616.53
7	DSR	New Reno	1500	10	42911.56
8	DSR	Sack	1500	10	41936.73
9	DSR	Tahoe	1500	10	42123.95

7. 7th case

a). Number of nodes 10
b). Packet Size 512

Now the numbers of nodes are taken 10 with again varying background traffic, DSR in case of 205312 bytes background traffic produces maximum throughput with Tahoe. The throughput is 47585.54 bytes.

8. 8th case

a). Number of nodes 10
b). Packet Size 1000

Once again with DSR produces maximum throughput with Tahoe in 376000 bytes background traffic. The throughput is 45616.53 bytes.

9. 9th case

a). Number of nodes 10
b). Packet Size 1500

Finally with 10 nodes and 564000 bytes background traffic DSR produces maximum throughput with New Reno which is 42911.56 bytes.

6 Summary

Now in 3 Nodes scenario, with background traffic of 79360 bytes, 155000 bytes and 232500 bytes and with three TCP algorithms (New Reno, Sack and Tahoe). It is found that AODV produces consistence throughput, where Reno and Tahoe produces high throughput with DSDV in the presence of 155000 bytes background traffic. SACK throughput is maximum with DSR in the presence of 232500 bytes as background traffic, while this is the maximum throughput in three node scenario recorded in the combination of SACK and DSR.

In five nodes scenario with background traffic of 79360, 155000 and 232500 bytes, NewReno produces maximum throughput with AODV, while SACK throughput is maximum in the presence of DSDV and DSR, but as whole the maximum throughput is come out in combination of SACK and DSR.

Finally, in 10 nodes scenario having the background traffic of 79360, 155000 and 232500 bytes, it is found that in the presence of AODV the throughput of all TCP variants is different and NewReno achieves the maximum throughput. In the presence of DSDV the maximum throughput is achieved by NewReno. While in Case of DSR the maximum throughput is gained y Tahoe.

7 References

[1] H. Zhou, “A Survey on Routing Protocols in MANETs ,” a technical report, March 2003.

[2] IEEE standard 802.11-2007, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, 2007

[3] Charles E. Perkins and Pravin Bhagwat, “Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers,” In Proc. of the SIGCOM 94 Conference on Communications Architecture, protocols and Applications, Augt 1994. Pp.234-44.

[4] C. E. Perkins, E. Belding-Royer, S. Das “Ad Hoc On Demand Distance Vector (AODV) Routing,” Request for Comments, 3561, July 2003.

[5] David B. Johnson and David A.Maltz, “Dynamic source routing in Ad hoc wireless networks”. In Mobile Computing, edited by Tomasz Imielinski and Hank Korth, chapter 5, pages 153-181. Kluwer Academic Publishers, 1996.

[6] V. Jacobson, “Congestion Avoidance and Control”, Proceedings of the ACM SIGCOMM', 1988, pp. 314-329.

[7] M. Allman, V. Paxson, and W. Stevens, “TCP Congestion Control,” Request for Comments, RFC 2581, Network Working Group, Internet Engineering Task Force, Apr. 1999

[8] S. Floyd, and T. Henderson, “The New Reno Modification to TCP Fast Recovery algorithm,” RFC 2582, Networking Group, Apr 1999.

[9] Matthew Mathis, Jamshid Mahadavi, Sally Floyd and Allyn Romanow, “TCP Selective Acknowledgement Options”, 1996.

[10] Kevin Fall, Kannan Varadhan “The *ns* Manual (formerly *ns* Notes and Documentation)” November 3, 2006.

BIOGRAPHY

Noor Mast (nm.afridi@hotmail.com) is currently a PhD candidate in the School of Engineering and Design, Brunel University, UK. He is also affiliated with the Institute of Information Technology, Kohat University of Science and Technology (KUST), Pakistan as a lecturer. His research interest includes TCP Analysis and Optimization in ad hoc wireless networks.

Kok-Keong Loo **(M’01)** a.k.a. Jonathan Loo received his MSc (Distinction) and PhD at University of Hertfordshire, UK in 1998 and 2003, respectively. Thereafter, he joined Brunel University, UK as a lecturer. Currently, he serves as a course director for MSc Digital Signal Processing and heads multimedia communications research group with 11 active PhD students. His current research interests include visual media processing and transmission, digital/wireless signal processing and wireless/broadband networks.