Communications on Applied Electronics (CAE) – ISSN : 2394-4714
Foundation of Computer Science FCS, New York, USA
Volume 5 – No.4, June 2016 – www.caeaccess.org

Different TCP Agents for Evaluating Performance of OLSR Routing Protocol in MANETS

Mubarak Mohammed Al-Ezzi Sufyan
MS(IT) Research Scholar
Institute of Business and Management Sciences
The University of Agriculture,
Peshawar-Pakistan

Khalid Saeed
Lecturer Computer Sciences
Institute of Business and Management Sciences
The University of Agriculture,
Peshawar-Pakistan

ABSTRACT

Transmission Control Protocol (TCP) is a transport layer protocol which uses three way handshake mechanisms for connection establishment. The research issue is to use TCP in MANETs for communication between source node and destination node. TCP has various agents, which are used for connection with source node or destination node. Seven agents for source and four agents for destination has been used in this research in order to evaluate the performance of OLSR routing protocol and to identify the behavior of OLSR over TCP's various agents by some performance metrics. The environment of this research is campus and the parameters for simulation are area size, packet size, simulation time etc in order to investigate the performance matrices such as sent rate, received rate and drop rate of data packets, packet delivery ratio, average end-to-end delay and average throughput by using NS-2.35 simulator. The results show that with all of the agents of the source node the performance of OLSR is better when the agent of the destination node is Agent/TCPSink/DelAck or Agent/TCPSink/Sack1/DelAck or Agent/TCPSink/Sack1.

Keywords

MANET, Routing, OLSR, MultiPoint Relays (MPR), TCP, FTP, Agents, Performance, Mobility.

1. INTRODUCTION

Mobile Ad Hoc Networks (MANETs) are temporary networks without any pre-established infrastructure and centralized administration. They are self-configurable (intelligent nodes), multi-hop and systematic devices communicate with each other within their communication range and also with those nodes which are outside their communication range with the help of hops by using route discovery mechanisms (connected by wireless link) anytime, anywhere with any one [1].

1.1 OLSR Routing Protocol

The Optimized Link State Routing (OLSR) protocol is a link-state proactive routing protocol, which was developed by INRIA (France) ["Jacquet", 2001] RFC 3626. OLSR operates as a proactive routing protocol (table-driven) of the classical link state algorithm that is a topology-based, because it exchanges information of the topology of the network with neighbor nodes and shares routing table information periodically. It is a better protocol as compared to other protocols, because it is used for large and dense networks and it works independently from other protocols [2, 3]. OLSR routing protocol has four types of control messages such as:

First: HELLO Message is generated by each node for neighbor's link sensing and MPR selection. It contains its own address and the list of its 1-hop neighbors by exchanging HELLO messages (HELLO INTERVAL is sent after every 2 seconds).

Second: Topology Control (TC) message is generated only by MPR nodes to advertise MPR selector information about the "topology" of the network by giving each neighbor selector sequence number incremented by one. It contains a list of the sender's MPR selector sent by TC messages periodically (TC message is sent after every 5 seconds).

Third: Multiple Interface Declaration (MID) message performs the task of declaring the presence of multiple interfaces on a node, because each node in the network has IP address, so it contains a list of the node's IP addresses in the network for transmitting these messages on more than one interface.

Fourth: Host and Network Association (HNA) message is used for association between OLSR network and other networks via node as a gateway, which is advertising OLSR network information to other networks.

1.2 MultiPoint Relay(MPR)

The responsibility of the MPR is to forward control traffic, declaring link state information and reducing the number of retransmission required by flooding control traffic to provide an efficient mechanism, so that each node selects its MPR set from among its 1-hop symmetric neighbors. The "MultiPoint Relay (MPR) set" for any node in the network is a set of nodes selected by that node from Symmetric 1-hop neighborhood, which retransmits its message to that node [2].

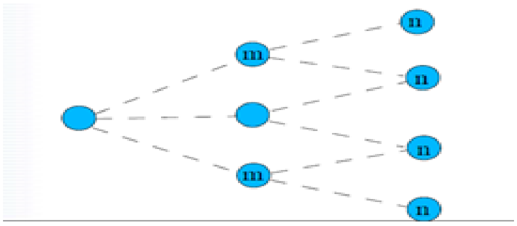

Figure 1: MPR Selection

1.3 Connection Pattern Models (Agents)

Agents represent endpoints where network-layer packets are generated or consumed, and are used in the implementation of protocols at various layers. It is used for the establishment of connection between the source node and the destination node. There are several agents supported in the simulator such as agents like TCP, SCTP, SRM, PLM UDP and DCCP. This research is based upon TCP agent.

Transmission Control Protocol (TCP) is a transport layer and connection-oriented protocol. It uses flow control, congestion control and error control mechanisms at the transport layer. TCP is needed to provide reliable end-to-end message transmission by receiving the ACK message from the destination node by the source node [4].

TCP connection agents are classified into two categories [5]:

First: The agent of sender (source node) is created by the following agents:

1) TCP Tahoe is based on Congestion Avoidance, Slow Start, and Fast Retransmit algorithms [6].
2) Agent/TCP/Reno is used for slow start, congestion avoidance and fast recovery of packets which are lost [6].
3) The Agent/TCP/Newreno is implemented to fix some fault of TCP-Reno by modifying the algorithm of fast recovery to dispose off Reno’s wait for a retransmit timer when multiple packets are lost from one window of data by sending sequence number that must be acknowledged before the fast recovery procedure is declared to be over [6, 7].
4) In [8] the researchers presented the Agent/TCP/Sack1 (Selective acknowledged) which is an extension to the algorithms of congestion control of the TCP Reno by making minimum changes to other congestion control algorithms such as Fast Recovery and Fast Retransmit.
5) Agent/TCP/Fack (Forward acknowledgment) is implemented to fix the drawbacks of TCP Reno with forward acknowledgment.
6) Agent/TCP/Linux is used for running TCP congestion controls modules for Linux Kernel.
7) Agent/TCP/Vegas is implemented which tries to detect the incipient stages of congestion before packet losses occur. It uses proactive mechanisms to increase and decrease the size of Congestion Window (CWND).

Second: The agent of receiver (destination node) is created by the following agents:

1. Agent/TCPSink is used with one ACK for each packet.
2. Agent/TCPSink/DelAck is used with configurable delay per ACK.
3. In [8] the researchers presented the Agent/TCPSink/Sack1 which is a selective ACK sink by combining it with a selective repeat retransmission policy, which can help to overcome these limitations. The receiving TCP sends back SACK packets to the sender informing the sender of data that has been received. The sender can then retransmit only the missing data segments.
4. Agent/TCPSink/Sack1/DelAck combines both the Sack1 mechanism with DelAck mechanism.

1.4 Traffic Pattern Model

Application traffic is used for generating data packets. There are two applications traffics commonly used with different TCP connections agents such as the following: File Transfer Protocol (FTP) and Telnet. In this research FTP is used.

FTP is an application layer protocol used to transfer computer files between the source node and the destination node on a computer network. FTP is built on client-server model architecture and uses separate control and data connections between the source node and the destination node [9].

1.5 Mobility Model

Mobility models are mathematical algorithms, which are designed in order to try modeling the behavior of real movement pattern of mobile users in geographic location. A mobility model is also used for performance evaluation in different simulations in order to represent the realistic movements of nodes. In this research Random WayPoint Mobility Model (RWPMM) is used which was first proposed by “Johnson and Maltz”. In this mobility model the mobile nodes move randomly and their location and speed changes over time. It is simple and widely available model, so it is the most popular mobility models to evaluate the performance of MANET routing protocol. Figure 2 shows the movement of each node from one waypoint Pi to the next Pi+1 [3, 10].

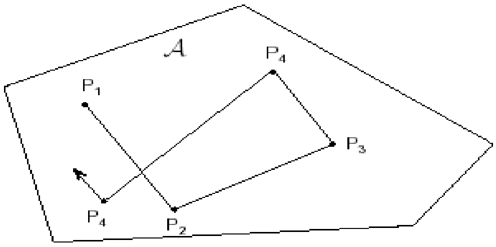

Figure 2: Node Movement in Random WayPoint Mobility Model

2. RELATED WORK

Kim *et al.* [11, 12] proposed TCP-Reno and TCP-Vegas connection for comparison between OLSR and AODV routing protocols in MANETs by using NS-2 simulator. In their result, over TCP-Reno and TCP-Vegas the throughput of OLSR is better as compared to the throughput of AODV, but over TCP-Vegas the performance of OLSR is better than the performance of AODV, also TCP-Vegas performs better for AODV, therefore TCP-Reno is more suitable for OLSR routing protocol.

Touati *et al.*, [13], investigated the performance of TCP Adaptive Retransmission TimeOut (RTO) in different MANET routing protocol, namely, OLSR, DSR and AODV by using NS-2 simulator to compare the throughput of TCP AR with TCP-New-Reno and TCP Westwood ABSE in order to propose an enhanced TCP called TCP Adaptive RTO (TCP AR), which is based on the values of different conditions of network. In their result, the performance of TCP AR is best as compared to the performance of TCP New Reno and TCP WestWood. Also the TCP AR improved the TCP throughput by using OLSR up to 161%, but by using DSR up to 277%.

Triantafyllidou *et al*., [14], presented an Optimized Scalable Simulation Model for TCP using OPNET simulator in order to evaluate the performance of OLSR routing protocol in MANETs.

Tahiliani *et al*., [15], evaluated the performance of DSDV, OLSR, DSR and AODV routing protocols in MANETs using TCP variants, namely, TCP-Tahoe, TCP-Reno, TCP-NewReno, TCP-SACK and TCP-Vegas by using NS-2 simulator. In their result, in general, TCP cannot distinguish between the packets lost by congestion and packets lost due to the link failures, but in all variants of TCP showed good results with DSDV, also TCP New-Reno and SACK TCP produced good results as compared to TCP-Tahoe and TCP-Reno with most of the routing protocols.

Liu *et al*., [16], analyzed the performance of TCP in MANET routing protocols such as OLSR and AODV by using NS-2 simulator. In their result, AODV performs better in maintaining TCP connections, also TCP congestion control mechanism did not work well over AODV, but OLSR shows good results with TCP in different mobility levels, also OLSR cannot route update interval when the node moves faster, therefore the TCP connection will break due to which the throughput of AODV is lower as compared to OLSR.

Morshed *et al*., [17], compared the performance of TCP variants, namely, TCP Tahoe, TCP Reno, TCP New-Reno and TCP Vegas over AODV, DSDV, DSR and OLSR by using NS-2 simulator. In their result, the comparison between the routing protocols, TCP-Tahoe and TCP-Reno are suitable for AODV, but TCP New-Reno is suitable for DSDV and TCP Vegas is suitable for DSR, while comparison between the performance on variants of TCP, the TCP Vegas showed highest efficiency and performs best as compared to other TCP agents.

Colizza *et al*., [18], proposed the performance of OLSR with Cross-Layer and TCP/UDP flows in MANETs by using OMNET++ simulator for studying the performance via ETX (Expected Transmission count) and PER (packet error rate) metrics. In their result, PER metrics enable better performance, when the traffic load supported by the network is large enough, also to enable support of QoS requirements.

Soni *et al*., [19], evaluated the performance of AODV, DSR, DSDV and OLSR routing protocols in MANETs via Multimedia traffic, namely, CBR Traffic and TCP Traffic by using NS-2 simulator. In their result, the performance of OLSR is better for TCP as compared for CBR, while AODV and DSR performed better for CBR traffic as compared to TCP traffic.

3. SIMULATION SETUP

In this research NS (Network Simulator) is used. This simulator is used for evaluating performance of OLSR routing protocol using different TCP Agents for connection between the source node and the destination node.

3.1 Network Simulation (NS-2.35)

Network Simulator (NS) was developed as part of the VINT (Virtual Internet Testbed) PROJECT. In 1995 NS-1 was developed, in 1996 NS-2 was developed, which than included object-oriented TCL in released from NS-1 to NS-2.35, and in NS-3. NS is an open source, which is used by the scientific community for network research as a discrete event simulator. It can be used in windows and Linux operating systems for simulating both wire and wireless networks. In this research NS-2.35 is used. It is available on "nsnam" website in (NS-allinone-2.35.tar.gz) zip file. There are numerous software, which are used with NS either included in NS or available separate. The software includes in NS are the following: NS for executing the TCL coding, NAM for showing the datagram of simulation, the trace file (.tr) is the outcomes from the simulation and are stored in it in different format such as a string, numbers, words and symbols, and Xgraph for showing the graphs of simulation from the trace-file [5].

3.2 Simulation and Analysis Software Tools

In additional to the NS-2.35 software tools, there are some separate software tools available, some of them need to configure with NS and another do not requires configuration. The software tools are used for building the scenario of the simulation and others are used for analysis and drawing the graph of the results obtained from simulation. The simulation and analysis software tools used in this research work are the following: TCL Script Language [20] for building the scenario, AWK is a particular program not included in NS also it does not need any configuration with NS, it used for generating the values of the performance metrics for example: throughput, End_To_End_Delay dependent on the event in the trace file such as: r="received packet", m="movement node" f="forwarded packet" and s="sent packet" etc [21] and Trace-Graph is a separate software tools Copyright (c) 2001-2005 by "Jaroslaw Malek", which is used for showing the information of the simulation in the form of graph, image and statistical format.

3.3 Configuring OLSR Packets in NS-2.35

OLSR routing protocol is not included in all versions of NS-2, therefore if anyone wants to use OLSR in NS must build OLSR packets in it by the UM-OLSR code and have to configure it. The UM_OLSR proposed by "Andrey Lyubimov" was chosen for this project as it is fully complied with IETF's "RFC 3626" and support all core functionalities of OLSR.

3.4 Parameters of Simulation

This section listed the parameters used for simulations in this research in the form of table as shown in table 1.

Table 1: Simulation Parameters

Operating system	Linux Mint 17.1 Cinnamon 64-bit
Simulator	NS-2.35
Routing Protocol	OLSR
Simulation Area	1500m x 1500m
Simulation time	30 seconds
Number of nodes	100
Connection type of source node	Agent/TCP Agent/TCP/Reno Agent/TCP/Newreno Agent/TCP/Sack1 Agent/TCP/Fack Agent/TCP/Linux

Connection type of destination node	Agent/TCPSink Agent/TCPSink/DelAck Agent/TCPSink/Sack1 Agent/TCPSink/Sack1/DelAck
Traffic type	FTP
Packet size	512 bytes
Node speed	5-10-15-20-25-30 m/s
Pause time	0.500000s
Mobility Model	Random WayPoint
Total number of scenario	24

3.5 Implementation Scenarios

There are many TCP agents dependent on the connection with the source node or with the destination node. This research used six TCP agents for connection with the source node, and four agents for connection with the destination node, therefore this research included twenty-four scenarios such as each one of TCP agent for connecting the source node with all the TCP agents of the destination node. The Table 2 shows all the required details.

Table 2: Scenarios of Simulation

TCP agents of Source	TCP agents of Destination	Scenario No
Agent/TCP	Agent/TCPSink Agent/TCPSink/DelAck Agent/TCPSink/Sack1 Agent/TCPSink/Sack1/DelAck	Scenarios number 1, 2, 3 and 4
Agent/TCP/Reno		Scenarios number 5, 6, 7 and 8
Agent/TCP/Newreno		Scenarios number 9, 10, 11 and 12
Agent/TCP/Sack1		Scenarios number 13, 14, 15 and 16
Agent/TCP/Fack		Scenarios number 17, 18, 19 and 20
Agent/TCP/Linux		Scenarios number 21, 22, 23 and 24

4. RESULTS AND DISSCUSION

This section of research paper includes the results for evaluating the performance of OLSR routing protocol. There are six main scenarios based on TCP agents of the source node, also there are four sub scenarios of each main scenario. The values of performance metrics are calculated from the trace file based on the event by using Xgraph, Trace Graph and the AWK language script. Also the graph of the performance metrics are generated by Xgraph, Trace Graph and Microsoft Excel.

4.1 Sent Rate

Sent Rate is the total number of data packets, which were sent by the source node.

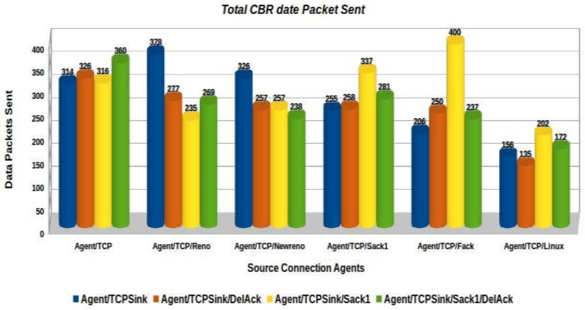

Figure 3: Sent Rate of data packet

Figure 3 shows the sent rate of data packets. At Agent/TCP for the source node connected with the Agent/TCPSink/Sack1/DelAck for the destination sent data packets more as compared to other agents. At Agent/TCP/Reno and Agent/TCP/Newreno for the source node connected with the Agent/TCPSink for the destination sent data packets more as compared to other agents. At Agent/TCP/Sack1, Agent/TCP/Fack and Agent/TCP/Linux for the source node connected with the Agent/TCPSink/Sack1 for the destination sent data packets more as compared to other agents.

4.2 Received Rate

Received Rate is the total number of data packets, which are received by the destination node.

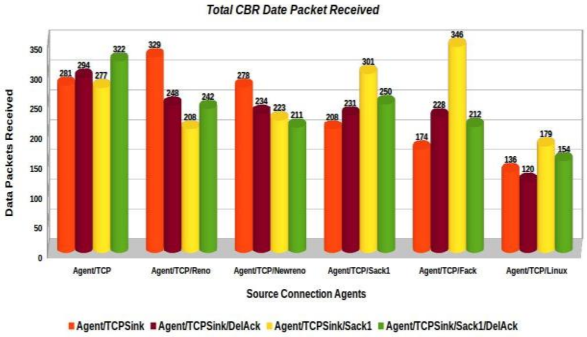

Figure 4: Received Rate of data packets.

Figure 4 shows the Received Rate of data packets. At Agent/TCP for the source node connected with the destination node at Agent/TCPSink/Sack1/DelAck, the destination node received data packets more than other agents. At Agent/TCP/Reno and Agent/TCP/Newreno for the source node connected with the destination node at Agent/TCPSink , the destination node received data packets more than other agents. At Agent/TCP/Sack1, Agent/TCP/Fack and Agent/TCP/Linux for the source node connected with the destination node at Agent/TCPSink/Sack1, the destination node received data packets more than other agents.

4.3 Drop Rate

Drop Rate is the total number of data packets, which were dropped by the destination node or any MPR nodes.

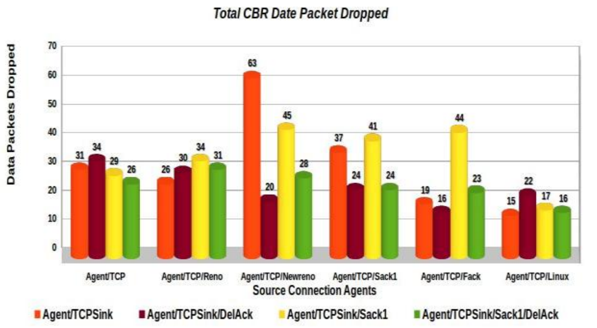

Figure 5: Drop Rate of data packet

Figure 5 shows the Drop Rate of data packets. At Agent/TCP for the source node connected with the destination at Agent/TCPSink/Sack1/DelAck, the OLSR network dropped less data packets as compared to other agents. At Agent/TCP/Reno for the source node connected with the destination at Agent/TCPSink, the OLSR network dropped less data packets as compared to other agents. At Agent/TCP/Newreno for the source node connected with the destination at Agent/TCPSink/DelAck, the OLSR network dropped less data packets as compared to other agents. At Agent/TCP/Sack1 for the source node connected with the destination at both Agent/TCPSink/DelAck and Agent/TCPSink/Sack1/DelAck, the OLSR network dropped less data packets as compared to other agents. At Agent/TCP/Fack for the source node connected with the destination at Agent/TCPSink/DelAck, the OLSR network dropped less data packets as compared to other agents. At Agent/TCP/Linux for the source node connected with the destination at Agent/TCPSink, the OLSR network dropped less data packets as compared to other agents.

4.4 Average End-To-End Delay

It is the average time taken by a data packet to reach the destination.

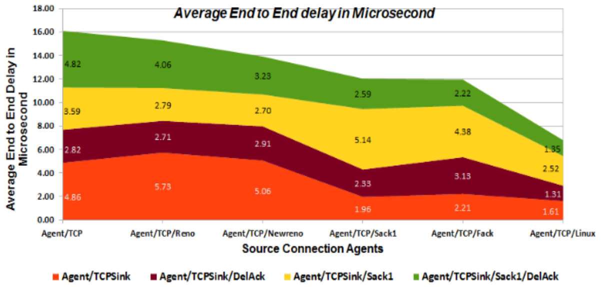

Figure 6: Average End to End Delay in microsecond.

Figure 6 shows the average End to End Delay in Microsecond. At Agent/TCP and Agent/TCP/Reno for the source node connected with the Agent/TCPSink/DelAck for the destination the end to end delay is less as compared to other agents. At Agent/TCP/Newreno for the source node connected with the Agent/TCPSink/Sack1 for the destination the end to end delay is less as compared to other agents. At Agent/TCP/Sack1 and Agent/TCP/Fack for the source node connected with the Agent/TCPSink for the destination the end to end delay is less as compared to other agents. At Agent/TCP/Linux for the source node connected with the Agent/TCPSink/DelAck for the destination the end to end delay is less as compared to other agents.

4.5 Packet Delivery Ratio

It is the ratio of the data packets successfully delivered to the destination divided by sent packets.

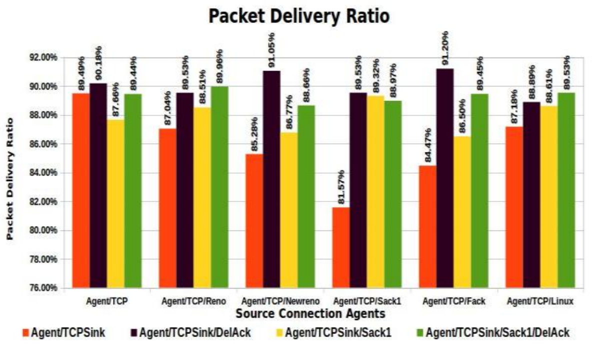

Figure 7: Packet Delivery Ratio of data packet

Figure 7 shows Packet Delivery Ratio of data packet. At Agent/TCPSink/DelAck for the destination delivered data packets more as compared to other agents when the agent of source node is Agent/TCP. At Agent/TCPSink/Sack1/DelAck for the destination delivered data packets more as compared to other agents when the agent of source node is Agent/TCP/Reno. At Agent/TCPSink/DelAck for the destination delivered data packets more as compared to other agents when the agent of source node is Agent/TCP/Newreno or Agent/TCP/Sack1 or Agent/TCP/Fack. At Agent/TCPSink/Sack1/DelAck for the destination delivered data packets more as compared to other agents when the agent of source node is Agent/TCP/Linux.

4.6 Average Throughput

An average throughput is the average rate of packets successfully transferred to their final destination per unit time. An average throughput is calculated as:

$$\text{Throughput (kbps)} = \frac{\text{Total packets delivered} \times \text{packet Size} \times (8/1000)}{\text{Total time of simulation}}$$

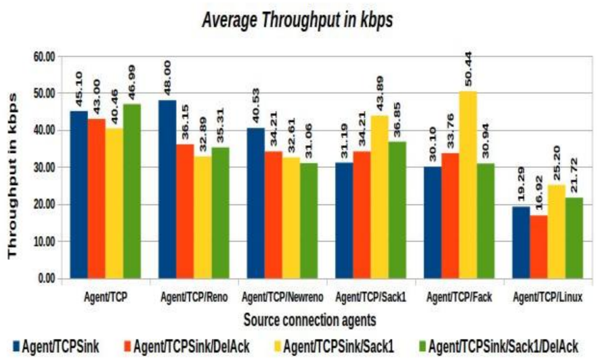

Figure 8: Throughput in kbps

Figure 8 shows throughput values in kbps. The average throughput of the OLSR network is more, when the agent of source node is Agent/TCP and the agent of destination node is Agent/TCPSink/Sack1/DelAck. The average throughput of the OLSR network is more, when the agent of source node is Agent/TCP/Reno or Agent/TCP/Newreno and the agent of destination node is Agent/TCPSink. The average throughput of the OLSR network is more, when the agent of source node is Agent/TCP/Sack1 or Agent/TCP/Fack and the agent of destination node is Agent/TCPSink/Sack1. The average throughput of the OLSR network is more, when the agent of source node is Agent/TCP/Linux and the agent of destination node is Agent/TCPSink/Sack1.

4.7 Path Length

Path length is the number of hops (MPR nodes) from a source to a destination for transmitting data packets between them. The average numbers of intermediate nodes (MPR) for the whole network. It is calculated by the average number of nodes receiving data packets and forwarding data packets between the source node and the destination node as the following:

4.7.1 An average number of nodes receiving packets

It is calculated as the sum of numbers of all the intermediate nodes (MPR nodes between source and destination nodes) receiving data packets sent by the source node / number of received data packets at the destination node.

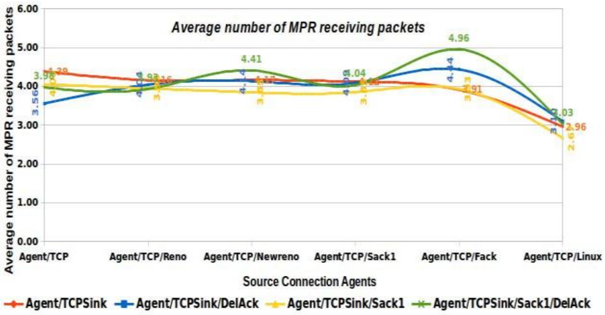

Figure 9: Average number of MPR nodes receiving data packets

Figure 9 shows the average number of MPR nodes receiving data packets. At Agent/TCP for the source node connected with the Agent/TCPSink/DelAck for the destination selected less nodes as MPR for receiving data packets as compared to other agents. At Agent/TCP/Reno for the source node connected with the Agent/TCPSink/Sack1/DelAck for the destination selected less nodes as MPR for receiving data packets as compared to other agents. At Agent/TCP/Newreno and Agent/TCP/Sack1 for the source node connected with the Agent/TCPSink/Sack1 for the destination selected less nodes as MPR for receiving data packets as compared to other agents. At Agent/TCP/Fack for the source node connected with the Agent/TCPSink for the destination selected less nodes as MPR for receiving data packets as compared to other agents. At Agent/TCP/Linux for the source node connected with the Agent/TCPSink/Sack1 for the destination selected less nodes as MPR for receiving data packets as compared to other agents.

4.7.2 An average number of nodes forwarding packets

It is calculated as the sum of numbers of all the intermediate nodes (MPR nodes between source and destination nodes) forwarding data packets sent by the source node / number of received data packets at the destination node.

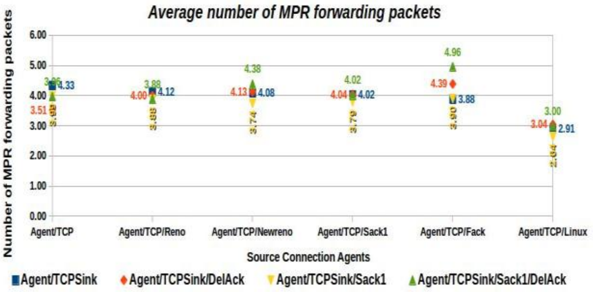

Figure 10: Average number of MPR nodes forwarding data packets

Figure 9 shows the average number of MPR nodes forwarding data packets. At Agent/TCP for the source node connected with the Agent/TCPSink/DelAck for the destination selected less nodes as MPR for forwarding data packets as compared to other agents. At Agent/TCP/Reno, Agent/TCP/Newreno and Agent/TCP/Sack1 for the source node connected with the Agent/TCPSink/Sack1 for the destination selected less nodes as MPR for forwarding data packets as compared to other agents. At Agent/TCP/Fack for the source node connected with the Agent/TCPSink for the destination selected less nodes as MPR for forwarding data packets as compared to other agents. At Agent/TCP/Linux for the source node connected with the Agent/TCPSink/Sack1 for the destination selected less nodes as MPR for forwarding data packets as compared to other agents.

5. CONCLUSION AND FUTURE WORK

5.1 Conclusion

This research provides an evaluation of the performance of OLSR routing protocol by using different TCP Agents for source node and destination node by using NS-2.35.

The performance of OLSR is evaluated based on the agent of the source node and the agent of the destination node by performance metrics as the following:

- The performance of OLSR is more efficient, when the connection of the source node is Agent/TCP and the connection of the destination is Agent/TCPSink/DelAck or Agent/TCPSink/Sack1/DelAck.
- The performance of OLSR is more efficient, when the connection of the source node is Agent/TCP/Reno and the connection of the destination is Agent/TCPSink or Agent/TCPSink/Sack1/DelAck or Agent/TCPSink/DelAck.
- The performance of OLSR is more efficient, when the connection of the source node is Agent/TCP/Newreno and the connection of the destination is Agent/TCPSink/DelAck or Agent/TCPSink/Sack1 or Agent/TCPSink.
- The performance of OLSR is more efficient, when the connection of the source node is Agent/TCP/Sack1 and the connection of the destination is Agent/TCPSink or Agent/TCPSink/DelAck or Agent/TCPSink/Sack1.

- The performance of OLSR is more efficient, when the connection of the source node is Agent/TCP/Fack and the connection of the destination is Agent/TCPSink or Agent/TCPSink/DelAck or Agent/TCPSink/Sack1.

In short, the performance of OLSR with all the agents of the source node is better when the agent of the destination node is Agent/TCPSink/Sack1/DelAck or Agent/TCPSink/DelAck or Agent/TCPSink/Sack1, because the maximum performance metrics produced good results under them.

5.2 Future Work

The following are the some research ideas identified while conducting this research:

- Evaluating the performance of OLSR routing protocol by using two-way TCP Agents (FullTcp).
- Developing a mechanism for TCP to distinguish between the packets lost by congestion and packets lost by the link failures.

6. REFERENCES

[1] Gupta, R., 2011. "Mobile adhoc network (MANETS): Proposed solution to security related issues. Indian Journal of Computer Science and Engneering (IJCSE), 2(5), pp.738-746.

[2] Clausen, T., Jacquet, P., Adjih, C., Laouiti, A., Minet, P., Muhlethaler, P., Qayyum, A. and Viennot, L. 2003. Optimized link state routing protocol (OLSR).

[3] Sufyan, M. M. A., Saeed, K. 2016, March Performance Evaluation of OLSR Routing Protocol using Different Mobility Models in MANETS. In International Journal of Computer Applications (IJCA) 137(8):23-30.

[4] Postel, J. Transmission Control Protocol. 1981.

[5] Fall, K. and Varadhan, K. 2005. The ns Manual (formerly ns Notes and Documentation). The VINT project, 47.

[6] Fall, K. and Floyd, S. 1995. Comparison of Tahoe, Reno and SACK TCP.

[7] Nishida, Y. 2012. The NewReno Modification to TCP's Fast Recovery Algorithm.

[8] Mathis, M., Mahdavi, J., Floyd, S. and Romanow, A. 1996. TCP selective acknowledgment options.

[9] Protocol, F.T., Postel, J. and Reynolds, J. 1985. File Transfer Protocol. Internet RFC 959, October.

[10] Yoon, J., Liu, M. and Noble, B. 2003, March. Random waypoint considered harmful. In INFOCOM 2003. twenty-second annual joint conference of the IEEE computer and communications. IEEE societies (Vol. 2, pp. 1312-1321). IEEE.

[11] Kim, D., Bae, H., Song, J. and Cano, J.C. 2005, June. Analysis of the interaction between TCP variants and routing protocols in MANETs. In Parallel Processing, 2005. ICPP 2005 Workshops. International Conference Workshops on (pp. 380-386). IEEE.

[12] Kim, D., Cano, J.C., Manzoni, P. and Toh, C.K. 2006, September. A comparison of the performance of TCP-Reno and TCP-Vegas over MANETs. In Wireless Communication Systems, 2006. ISWCS'06. 3rd International Symposium on (pp. 495-499). IEEE.

[13] Touati, H., Lengliz, I. and Kamoun, F. 2007, September. Performance of TCP Adaptive RTO in ad-hoc networks based on different routing protocols. In Mobile Wireless Communications Networks, 2007 9th IFIP International Conference on (pp. 176-180). IEEE.

[14] Triantafyllidou, D. and Agha, K.A. 2007, October. Evaluation of TCP performance in MANETs using an optimized scalable simulation model. In Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2007. MASCOTS'07. 15th International Symposium on (pp. 31-37). IEEE.

[15] Tahiliani, M.P., Shet, K.C. and Basavaraju, T.G. 2010, September. Performance evaluation of TCP variants over routing protocols in multi-hop wireless networks. In Computer and Communication Technology (ICCCT), 2010 International Conference on (pp. 387-392). IEEE.

[16] Liu, P.C., Chen, D.Y., Hu, C.L., Sun, W.C., Lee, J.H., Chou, C.K. and Shih, W.K. 2011, February. Analyzing the TCP performance on mobile ad-hoc networks. In Advanced Communication Technology (ICACT), 2011 13th International Conference on (pp. 143-148). IEEE.

[17] Morshed, M.M., Rahman, M.U., Rahman, M.H. and Islaml, M.R. 2012, May. Performance comparison of TCP variants over AODV, DSDV, DSR, OLSR in NS-2. In Informatics, Electronics & Vision (ICIEV), 2012 International Conference on (pp. 1069-1074). IEEE.

[18] Colizza, M., Pratesi, M., Santucci, F., Pollio, S. and Rigazzi, G. 2012, September. Performance of OLSR in MANETs with Cross-Layer Metrics and TCP/UDP Flows. In Wireless Communications, Networking and Mobile Computing (WiCOM), 2012 8th International Conference on (pp. 1-5). IEEE.

[19] Soni, S.J. and Shah, J.S. 2015, April. Evaluating Performance of OLSR Routing Protocol for Multimedia Traffic in MANET Using NS2. In Communication Systems and Network Technologies (CSNT), 2015 Fifth International Conference on (pp. 225-229). IEEE.

[20] Ousterhout, J.K. 1993, TCL and the TK Toolkit. Computer Science Division, Department of Electrical Engineering and Computer Science, University of California, Berkeley.

[21] Close, D.B., Robbins, A.D., Rubin, P.H., Stallman, R. and van Oostrum, P. 1995. The AWK Manual.