***I.J. Wireless and Microwave Technologies*, 2018, 2, 52-63**
Published Online March 2018 in MECS(http://www.mecs-press.net)
DOI: 10.5815/ijwmt.2018.02.05

***Available online at* http://www.mecs-press.net/ijwmt**

MANET Testbed using Raspberry PIs

Salem Sati[a,*], Ahmed El-bareg[b]

[a]Misuratat University, Highway road, Misurata 218512, Libya
[b]The Collage of Industrial Technology, Ring-road 3, Misurata 218229, Libya

Received: 09 October 2017; Accepted: 17 January 2018; Published: 08 March 2018

Abstract

Mobile ad-hoc networks (MANETs) are without any infrastructure and consists only of equal peers. This paper presents an empirical tests for MANET routing protocols. The testbed based on Linux platform installed in mobile computers. Some parameters are analyzed such as movement speed and the number of hops which impact on routing performance. The testbed consists of several Raspberry Pis (RPis) without the need for any central master device. For making the RPis mobile and independent of any fixed power sockets, each of them is powered by a battery. On the other hand, for the evaluation of the testbed, two routing protocols are chosen. The first protocol is called BABEL, which considered as a distance vector routing algorithm. The other one, the Optimized Link State Routing Protocol (OLSR) which considered as proactive routing protocol. The use case was a multi-hop download of files with different size. The aim is to evaluate how multiple hops influence the bandwidth and delay. The results show that OLSR performs better regarding the throughput. But Babel has less delay and faster regarding convergence.

Index Terms: MANET Routing, MANET Test-bed, Raspberry PIs, Performance Comparison, Multi-hop Influences.

1. Introduction

Wireless communication is an essential technology today to connect computers with each other. Most of the connected mobile devices use infrastructure mode. This mode consists of a master device such as wireless access point. This infrastructure is expensive to build and maintain. Especially in sparsely regions with a few of users and limited radio range. Furthermore, the infrastructure mode is vulnerable regarding the topology and mobility in the bounded area. In such zones, a method of communicate without the existence of infrastructure

* Corresponding author. Tel.:
E-mail address:salem.sati@it.misuratau.edu.ly

mode is required.

This challenge may solved by mobile ad-hoc networks (MANETs). Ad-hoc networks consist of nodes collection which is connected without any central master device. Each node in MANET communicates directly with another node without the need for the master device. MANET nodes organize themselves dynamically. The routing functionality in MANETs is based on source routing fashion. For communicating with nodes out of radio range, the node uses multi-hop routing. The nodes on the path serve as a router and relay the packet to next hop. The network topology dynamically changes over time when nodes change their position. In general, MANETs routing protocols are commonly classified into three types which are proactive, reactive and hybrid routing protocols. For proactive which also known as table-driven or distance vector routing protocols, each mobile node has to collect routing information for all destinations in the network topology. The node keeps this information fresh by exchanging route updates with other nodes in the network. MANET proactive routing protocols such as Destination-Sequenced Distance-Vector (DSDV), Optimized Link State Routing Protocol (OLSR) and Babel. The second type of MANET routing, a reactive routing protocol. The path of this routing is discovered when a node needs to send a packet to a specific destination. As a result of this mechanism, this kind of protocols generate less routing overhead but it suffering from the delay. Reactive MANET routing include protocols such as Dynamic Source Routing (DSR) and Ad-Hoc On-Demand Distance Vector (AODV). Finally, the third type of MANET routing is Hybrid routing protocols, which is a combination of proactive and reactive properties. Hybrid protocols such as Zone Routing Protocol (ZRP).

2. Related Work

There are many studies related to the evaluation of ad-hoc network routing protocols. Several of those evaluation conducted with testbed method by mobile nodes such as a laptop or Raspberry Pis. There are two main approaches for routing evaluation. first, by use simulators such as in paper [1]. The simulator used for collect information about the network. The second approach is using real testbeds. But most researchers consider only one protocol for evaluation such as in [2]. The authors use two different test situations. In both cases five laptops creates an ad-hoc network. The authors show that the traffic decreases if the source node moves away from the destination. In paper [3] a real testbed is used to evaluate the routing protocol. The authors use only OLSR routing protocol. Furthermore, the authors of the paper [4] which suggest the implementation and experiment investigation of OLSR routing protocol. This evaluation conducted based on TL-MR3040 routers installed with the OLSR protocol. Results of the paper show that the OLSR performance is good. The authors of the paper [5], they investigate the efficiency of OLSR routing protocol by conducting testbed. Evaluation of the paper was deeply analyze ad-hoc routing performance in certain conditions. Ad hoc nodes implemented by using Raspberry Pi with USB wireless network card model of TP-Link WN722N. In the paper [6], the authors present the required steps to implement and configure an inexpensive testbed. This testbed built using Raspberry Pi nodes for communications and storage. This testbed considers the network coding capabilities. This testbed can be used for any applications experiments. The paper [7] demonstrates the feasibility of a mobile mesh network testbed. The paper performs measurement experiments in an indoor environment. The testbed has Linux as the operating system. Furthermore, the testbed uses OLSR routing protocol to enable routing and delivering of the packets. The mobile nodes are Raspberry Pi computers. The paper conducting benchmark measurement and then comparing the results of a different mobility models. The authors of the paper [8] show deep analysis of three MANET routing protocols, which are; AODV, DSDV and OLSR. Evaluation conducted using NS-3 by changing node density. This paper concludes that OLSR has best performance compared with other two routing protocols. The paper [9] develops a prototype of real-time monitoring and motion detection. Most of sensors forming by the Raspberry Pi board, This prototype transferred the data via Wi-Fi/ 802.11 or via wired connectivity through the closest switch. In the paper [10], authors propose comparative simulation study of two proactive protocols. These protocols are OLSR and DSDV. Simulations conducted using (NS-2.35). The numerical results show that, OLSR has performance better than DSDV protocol.

3. Testbed

The testbed used in this paper consist of Raspberry Pis (RPis) [11] computers. All RPis are model B+ Version 1.2. RPis run Weezy Raspbian Linux of the version 3.18.7+ #755 armv61 GNU/Linux. The network card was a WLAN 802.11n which is USB Adapter in Nano Size. This WLAN has RT5370 chipset. This adapter supports the 802.11b/g which has throughput up to 27 Mbps. Each WLAN has a static private IP address for connecting with the ad-hoc. This ad-hoc has the network address of 192.168.178.0\24. All RPis use their interface wlan0 to create a wireless ad-hoc network. RPi 1 has a special role because it is configured as an internet gateway as shown in Fig 2. To connect the device physically to the internet, the LAN connection configured via eth0 of RPi 1. In order to work as a gateway, the routing table has to be extended manually. The first step is to allow forwarding between interfaces. After that, all traffic which comes out of the MANET over wlan0 will be forwarded to eth0. But in the opposite direction, only packets which belong to an established or related connection will be forwarded to the MANET. The first connection test run with fixed positions. Power batteries are added to the RPis to make them as mobile devices. Each battery has a charge of 2200 mAh and powers devices as shown in Fig.1.

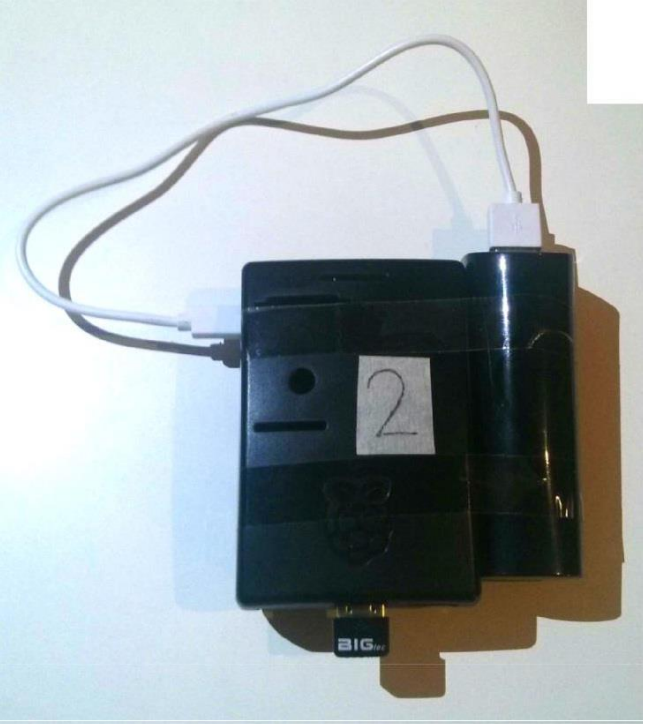

Fig.1. Raspberry Pi Powered by Battery

The controlling of the transmission range conducted by limiting the WLAN energy of each RPis, A topology visualization is presented in Fig 2 which is as cascaded or line topology. The RPi 1 which is configured as gateway as shown right on the top of Fig.2. The first RPi is connected via eth0 to the internet and shares its internet connection to the ad-hoc network via wlan0. The second RPi is in other floor level downstairs of university building, the third floor level has the third RPi and so on. An other test take place at the university as shown in Fig.3. The first RPi is placed in an office and wired to the internet connection. There are other nodes which are placed in the corridors with open space contact of neighbour. The third and main test scenario was on the campus between complexes of different buildings. The first RPi is placed on the balcony of the first floor. The next RPi is placed on outside of the next building on the ground floor. Each RPi's position is marked in the range of RPis 2 to RPi 7. Each test of three scenarios start with a 1-hop connection between RPi 1 and the next RPi. At first, the download speed with 1 hop is tested with several files. The files size of 10kb, 100kb, 500kb, 1mb, 10mb, 20mb and 1gb. We deploy different sizes for test different traffic loads. The file content consists of random binary data. To control the specific RPi, it has a second wireless interface which can be accessed via SSH.

4. Routing Protocols

The two routing protocols evaluated in this paper are proactive protocols. These routing protocols called OLSR [12] and Babel [13] routing protocols.

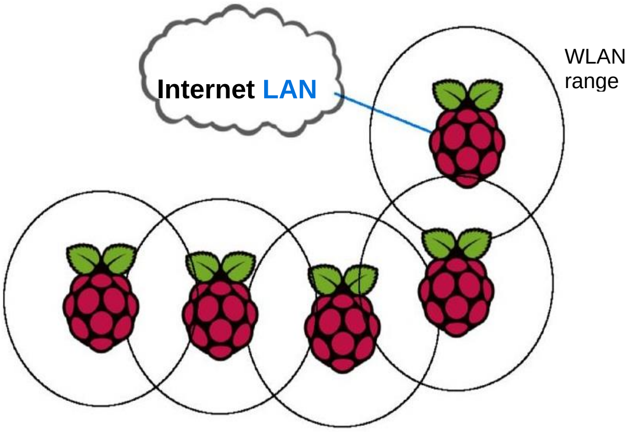

Fig.2. Raspberry Pis loosely Connected via Wireless lan

In our evaluation, we consider throughput analysis based on the ad-hoc network. We performed an empirical evaluation where we deploy multiple traffic loads on the network. Also, we consider the number of hops as a parameter of the link quality. Moreover, we investigate the impact of multiple hops on throughput and delay.

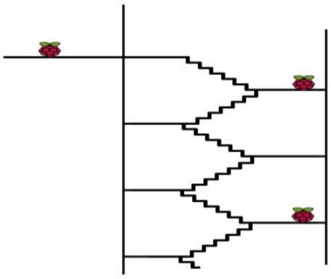

Fig.3. Raspberry in Different Positions

4.1. Babel

Babel is a routing protocol for mesh networks. It was developed at the University Paris-Diderot. Babel finds shortest paths with avoiding loops in this paths, Babel supports wired and wireless networks. Furthermore, Babel has the ability to address with IPv4 and IPv6. If the network is a highly dynamic topology, Babel rebuilds the network graph by routes updates. Routing loops may occur but loops will be removed by Babel convergences. This convergence takes a long time in special scenarios. Babel convergence depends on the network size and Babel scalability overhead, but in most scenarios convergence time measured in minutes. Babel uses many of control message types to maintain the network graph. The HELLO message is a very important control message type. It is used to discover neighbours and determining link costs. HELLO messages are distributed based on multicast manner in the network. Nodes which receive this message, answers with the IHU message. The IHU will be sent as a unicast message to the sender of the HELLO message. IHU is called the “I hear you“ message. IHU message shows that a link can be used in a bi-directional connectivity. The link cost is transmitted to the HELLO sender with the IHU message. Each Babel node sends the HELLO message periodically to all of its neighbours even the network topology doesn’t change. Every node answers every HELLO message with an IHU message. This causes an overhead of protocol messages. An interesting feature of Babel is that it is able to connect devices with different IP address versions. Furthermore, when the node moving dynamically leads to a shorter connectivity duration. Therefore, If the count of correct received HELLO messages decreases below a threshold, the link will be marked with infinite costs. Depending on the configuration Babel use different information to calculate the routing tree for each node. The cost of a route is the sum of all hops between source and destination. This cost called routing metric. The metric of the route/path has to raise strictly with each hop. Babel uses Bellman-Ford algorithm to find the shortest path. Although the Bellman-Ford algorithm can handle negative costs, Babel considers only positive costs to forward user content over multiple hops, Babel selects the route depending on information about the hops. For example routes with small metrics will be preferred. Links to stable neighbours will be preferred too. The Bellman-Ford algorithm is used by Babel for the route calculation.

- Bellman-Ford algorithm: The Bellman-Ford algorithm is an algorithm to calculate the costs from one node to all destinations. The Bellman-Ford algorithm is an algorithm to calculate the costs from one node to all other nodes. In contrast to the Dijkstra algorithm, the Bellman-Ford algorithm can handle negative edge costs. Edges are part of unidirectional links. Bidirectional edges consist of two links with opposite direction. Costs of a path are the sum of the edges to reach this destination node from the source node. The source node starts with cost value equal to 0. Because the node must reach itself, where there is no edge has to be used. All other nodes have infinite costs as an initial value. Now the algorithm runs several phases to determine better costs to the destination. If the graph has (n) edges, then the algorithm uses (n-1) phases. A path, which uses more edges would visit one node more than one time and create a loop. In each phase, the algorithm checks all edges and calculates the cost of the corresponding final node. Therefore the sum of cost for the first node and the edge is calculated. If the new value is lower than the old one of the path, the old value will be replaced with newly calculated path cost. In every phase (i) the algorithm should detect the shortest path with (i) hops. At the end, after (n-1) phases, the algorithm checks whether there are loops with negative costs. This is done by considering the sum of the first node cost added to edge costs compared with the cost of the final node cost in the path. If the final node in the path has a higher cost, this means that a negative loop is found. A negative loop causes that no minimal route can be found for the shortest path. The algorithm has a complexity of O(n,m) with (n) nodes and (m) edges. It has to be executed on each node which causes an complexity of $O(n^2,m)$.
- Routing table updates: After a Babel node has computed its routing tree, it advertises its routing table to its neighbours. Then the node compares their own information with the new route updates to refresh their local table if necessary. Babel exchanges route updates as TLV records. Babel control message includes several TLVs records, Babel control message consist of many tables such as Interface Table, Neighbour

Table, Source Table and Route Table.

4.2. OLSR

The Optimized Link State Routing Protocol (OLSR) [14] is one of routing protocols in MANETs. OLSR is designed for minimizing the control traffic of proactive routing updates. OLSR routing uses source routing fashion. OLSR is a proactive routing protocol which means that possible routes are calculated for all nodes in the network graph. In opposite to reactive protocols which depends on route discovery of the failed path. But proactive routing causes an overhead because all routes are distributed to all neighbors. The node uses HELLO messages periodically to detect its 1-hop and 2-hop neighbors. The answer form the directly connected node as HELLO message contains a list of 1-hop neighbors of the node. The directly connected node is a neighbor if there a direct symmetric link between both nodes. To reduce overhead generated by the node as broadcast control messages, there are nodes called Multi Point Relays (MPRs). These nodes responsible for forwarding broadcasts of control messages. Each node chooses its own set of MPRs out of its 1-hop neighbors. The MPRs are selected to reach all 2-hop neighbors with minimum hops and links. The other neighbor nodes receive and process the broadcast messages, but they don't retransmit them. In this manner, every broadcast message is sent to all nodes with a minimum overhead. The set of MPRs can change with the re-transformation of the network if some devices are changing their positions. Each MPR maintain a list of nodes which uses this node as a MPR. Additionally, the MPR sends topology-control (TC) messages to distribute information of possible links and routes. Based upon these TC messages, each node creates or updates its routing table. Functions of OLSR can be divided into a core and an auxiliary part. The core is mandatory for the working of OLSR. The auxiliary functions support the core or offer additional services. The core aggregates function for packet format and forwarding, neighbor detection, MPR selection, topology control message diffusion and route calculation. The auxiliary functions offer for example the connection to non-OLSR interfaces. A non-OLSR interface is a network interface of a node which doesn't participate in OLSR but offers, for example, an internet connection. To create a connection to a local network or the internet, OLSR uses Host and Network Association (HNA) messages. These announce show that a node has the connection to a given network. IPv6 is supported by OLSR, too. Since 2015 OLSR will not be improved furthermore. It will be replaced by OLSRv2. This paper uses OLSR for evaluation using the testbed.

5. Testbed Software

Some software needed for preparing of MANET testbed nodes using the Raspberry Pi boards. This process needs installing and set up an operating system and applications. Regarding the operating system used in MANET nodes was Raspbian, this platform developed based on Debian kernel. Raspbian used to manage Raspberry Pi hardware resources. Moreover, Raspbian is considered more than the operating system. This operating system includes many extra software packages for making software installation of Raspberry Pi more easier and flexible. For preparing and setting up the testbed of MANET, In addition to the operating system installation, some software packages also needed to upload in Raspberry Pi. These packages such as OLSR and Babel routing protocols which used for evaluation of MANET routing protocols. Both routing protocols mentioned in this paper has different configuration.

5.1. Babel Configuration

Babel is installed on the RPis using the package manager. This package which called Raspbian is used because it offers a stable and easy method for installing Babel daemon. Before starting the daemon of the gateway RPi 1, RPi 1 configuration file edited for modifying some parameters such as IP address, SSID, and channel frequency. Furthermore, the RPi 1 announces that it has an internet connection. To do that, the line ”redistribute it 0.0.0.0/0 le 0 metric 128” should be added to the configuration file. Now Babel started with the command “babeld wlan0 -d 1”. wlan0 is the wireless network interface, Babel should run with the option -d 1, This option debug and output the results on the terminal or shell. This debug option shows the known neighbors for each node.

5.2. OLSR Configuration

OLSR is also installed via the package manager. We install OLSR for the evaluation and comparison with Babel. The configuration file of the wired Ethernet interface of RPi 1 was modified. The option HNA must be activated. This activation of HNA for announces an internet gateway, the line “0.0.0.0 0.0.0.0“added to the HNA4 block. The first four zero describes the network address, and the following four zeros are the network mask. For IPv6 the line”: 0“should be added to the HNA6 block. With these settings, OLSR can be started with the command” olsrd - I wlan0 -d 1“. In contrast to Babel, we have to specify the interface with the parameter -i. The parameter -d 1 enables the debug output of the 1-hop and 2-hop neighbors and their costs.

6. Experimental Results

We connect MANET network for evaluation and comparing between OLSR and Babel routing protocols, for this evaluation, we need to define and use the following metrics:-

- Transmission range: the Wi-Fi transmission range of the nodes is controlled, due to the loss of energy and connectivity issues. The range control conducted by limiting the transmitted signal. Therefore, it is very important to investigate this metric which related to node connectivity in MANET.
- Connectivity: by assuming that the nodes are a move in particular area, the network graph will change with the time. Knowing these properties of the links and its graph in MANETs helps that the nodes are connected or not, this will allow us to prepare MANET routing to determine paths with a number of hops.
- Throughput: In the context of MANETs, throughput is the rate of the message delivered over a link or path and it is essentially measured in the unit of bits per second (bps).

To reflect the tests of different scenarios, we set up MANT by the RPIs using a WiFi-USB network interface card. Then we deploy specific experiments to obtain the above-mentioned metrics. Following is the description of the different scenarios that we used. The first scenario which includes a little experiment. This scenario shows that the range controlled by the transmitted energy of wireless network interface card. The experiment was done by connecting RPi 1 with two hops to reach RPi 3. The testbed was tested to check the connectivity and number of hops in addition to the throughput. In the simplest scenario environment, only OLSR was tested and evaluated. Table 1 shows that the first hop works great and offers a good throughput ranging from 570 to 650 kbps. But in the second hop, the bandwidth drops down to about 15 percent of the single hop throughput. The packet loss and overhead increases rapidly. In the third hop, nearly no connection could be held. Downloading the 10 and 100 kb test file was successful after a lot of trials. A packet loss of 85 percent was measured. The main problem is that when signal strength decreases leads to throughput decreases also. The link quality was quite good. Although different positions for each RPi are used, no enhancement was observed. Even if each link has a minimum distance between two RPis, Therefore, there is no reliable communication between nodes when the number of hops more than 3 hops. This is caused by the obstacles between nodes. Which leads to downloading files with minimum bit rate speed. The other test environment two which is located outdoors between two building complexes demonstrate the best results. Here the testbed used up to 7 RPis in a row. In maximum, a 6-hop routing can be evaluated as shown in Table.2. The first challenge was to figure out the optimal locations for each RPi. Again, we have a visible connection for each link between two neighbouring RPis. In general, the testbed was handled by small groups of students. The measurement for OLSR was taken by considering the throughput and delay with starting the OLSR daemon, our experiment conducted when most students are in the university during the lectures time.

Table 1. OLSR Environment 2

Hop count	file size	bandwidth kb/s	time min:sec
1	10kb	654	00:00,02
1	100kb	535	00:00,20
1	500kb	552	00:00,90
1	1mb	571	00:01,80
1	10mb	572	00:18,00
1	20mb	560	00:37,00
1	100mb	570	03:10,00
1	1gb	570	30:40,00
2	10kb	35	00:00,30
2	100kb	40,1	00:02,50
2	500kb	31,3	00:16,00
2	1mb	48,4	00:21,00
2	10mb	44,7	03:49,00
2	20mb	67	05:06,00
3	10kb	26,4	00:00,40
3	100kb	9,8	00:11,00

For the sparse environment as shown in table 2. The 1-hop connection was not affected by obstacles such as students bodies. The 2-hop connection is slightly affected. The 100mb and 1gb test wasn't successfully done because they take a long time. The count of hops increases leads to the throughput decreases. RPi 7 was placed far from the RPi 1 with 6 hops. With each increased hop from the traffic generator, the throughput convergences to zero. Tests with an eighth RPi was not successful.

Nearly 95 percent of packet loss of downloaded file. Downloading the 10mb file takes 6 minutes and 54 seconds. The update of the routing table of OLSR takes much time. This means that at each position, we have to pause walking for updating the routing table. After these moves, the 10 mb file was downloaded completely. The tests for Babel routing protocol are done in the evening. At this time, nearly all students are at home. This leads to best connectivity with lower bandwidth within each increased hop. But just like OLSR, the bandwidth decreases with each increased number of hops. The measured values are presented in Table 3. The results show that the bandwidth is nearly halved at each hop. An exception is between the first and second hop, here the reduction of the throughput roughly down to one fifth.

Table 2. OLSR Environment 3

hopcount	file size	bandwidth kb/s	time min:sec
1	10kb	554	00:00,02
1	100kb	535	00:00,20
1	500kb	560	00:00,82
1	1mb	567	00:01,90
1	10mb	576	00:18,00
1	20mb	567	00:35,00
1	100mb	558	03:13,00
1	1gb	579	29:54,00
2	10kb	235	00:00,04
2	100kb	78,6	00:01,30
2	500kb	75,3	00:06,60
2	1mb	81,7	00:13,00
2	10mb	84,6	02:01,00
2	20mb	90,4	03:47,00
3	10kb	20,8	00:00,50
3	100kb	11,2	00:19,00
3	500kb	16,6	00:45,00
4	10kb	12,3	00:00,80
4	100kb	7,54	00:13,00
4	500kb	33,2	00:21,00
4	1mb	27,5	00:37,00
4	10mb	19,5	11:36,00
5	10kb	4,34	00:08,70
5	100kb	3,98	01:04,00
5	500kb	10,1	04:38,00
6	10kb	9,2	00:01,10

Finally the throughput experiment was performed in different environment. The number of hops controlled by routing protocols of the establishment paths between the nodes in the MANET.

We had to put the nodes in the topology of Figure 2 to assure that OLSR routing protocol built the routing table in every MANET node. The way we needed it, that is, RPsi connected by one-hop to next RPi1 and by multi-hops to the traffic destination.

Table 3. Babel Environment 3

hopcount	file size	bandwidth kb/s	time min:sec
1	10kb	552	00:00,02
1	100kb	542	00:00,20
1	500kb	565	00:00,90
1	1mb	545	00:01,90
1	10mb	572	00:18,00
1	20mb	568	00:36:00
1	100mb	572	03:10,00
1	1gb	567	29:24,00
2	10kb	84,5	00:00,12
2	100kb	79,9	00:01,20
2	500kb	91,4	00:06,10
2	1mb	75,2	00:13,30
2	10mb	82,3	02:02,00
2	20mb	84,1	03:35,00
2	100mb	90,3	18:26,00
3	10kb	60,4	00:00,16
3	100kb	45,3	00:02,20
3	500kb	50,1	00:09,90
3	1mb	44,8	00:22,30
3	10mb	56,6	02:56,00
4	10kb	23,5	00:00,40
4	100kb	26,2	00:03,80
4	500kb	9,1	00:53,00
4	1mb	21,8	00:45,60
4	10mb	26,8	02:45,60
5	10kb	15,2	00:00,65
5	100kb	9,8	00:10,20
5	500kb	10,1	00:51,20
5	1mb	5,6	02:58,00
6	10kb	5,7	00:01,80
6	100kb	4,6	00:21,70

Clearly using our results we demonstrated that there is a strong relationship between packet loss and movement speed. Obviously, at speeds of around 1.5 m/s Babel would become insufficient as packet loss approach 100% of the loss. Using the results we show that number of hops will also cause more throughput degrading. We also observed from our results that most of the increase in packet loss is coming from node mobility speed. This paper shows that while Babel routing protocol is appropriate for low-speed MANTsAs mentioned in Tables 2 and 3, we observed that OLSR gives high traffic throughput compared with Babel. This comes from that Babel generate more control traffic compared with OLSR. Babel update problem comes from the mechanism of Babel depending on route discovery. The idea of Babel is to send the topology updated in a different period of time depending on the topology changes. This behavior of Babel causing more network overhead, Regarding OLSR this routing protocol sends the route update in the periodic interval. this behavior leads to constant control traffic overhead. In addition, OLSR use MPR which works as cluster head to optimize the link state mechanism, therefore, Babel considered as fast convergence proactive protocol compared with OLSR. where Babel collects and updates the information of the network periodically and when graph changes. Therefore, OLSR takes double time of Babel for convergence.

7. Conclusions

This paper contributes a real evaluation of MANET routing protocols for delivering the data and finding the best path. We analysis the throughput, performance, and stability of two distance vector routing protocols which are OLSR and Babel. These two proactive routing protocols are commonly used in MANET, through real evaluation and experimentation using Raspberry PIs computers. Our evaluation and experiments are based on a testbed with various node mobility conditions at different traffic loads. Our different scenarios with different mobility and traffic loads have provided detailed results that compare the performance of the two routing protocols. Obviously, we can observe that Babel is considered as the most lightweight protocol regarding of consuming hardware resources such as memory and processing. Moreover, if the comparison is related to large-scale or dense wireless networks with multi-hop routing such as in our environment of 7 RPi's, then the routing update mechanism determines how to update topology changes by sending control traffic as routing overhead. In such scenarios, OLSR considered better than Babel, This comes from that OLSR has its strictly periodical interval for sending topology dynamics and routing update, where Babel consider the route update also an event update. This behaviour of Babel lead to that Babel in terms of link stability, and even convergence capabilities are better than OLSR. But these Babel extra route update and control traffic have the disadvantage of high overhead of routing protocol. Regarding the throughput metric OLSR protocol considered as better than Babel especially in the large scale and dense networks, in these networks as the number of hops increases in the path the throughput decreases with about 15 percentage. For future work, we plan to extend our testbed by considering more MANET routing protocols, Furthermore, we would like to take into account deep investigation of the comparison between proactive and reactive routing protocols.

References

[1] J. S. G. Ramandeep Kaur Nagra and G. S. Grewal, “Simulation based Analysis of AODV, BABEL and PUMA Protocols for Adhoc Network,” in International Journal of Computer Application, 2012.

[2] M. I. Elis Kulla, Masahiro Hiyama and L. Barolli, “Comparison of Experimental Results of a MANET Testbed in Different Environments Considering BATMAN Protocol,” in Third International Conference on Intelligent Networking and Collaborative Systems, 2011.

[3] F. X. Leonard Barolli, Makoto Ikeda and A. Duresi, “A Testbed for MANETs: Implementation, Experiences and Learned Lessons,” IEEE Systems Journal, vol. 4, no. 2, 2010.

[4] F.Azizan, M. Zuhairi and H. Salleh, “Optimised Link State Routing Protocol Testbed and Performance Evaluation”, International Journal of Computer Science and Network Security(IJCSNS), vol. 16, no. 10, 2016.

[5] D. Lumbantoruan and A. Sagala, “PERFORMANCE EVALUATION OF OLSR ROUTING PROTOCOL IN AD HOC NETWORK”, ARPN Journal of Engineering and Applied Sciences , vol. 10, no. 3, 2015.

[6] C. Sørensen, N. Hernández and J. Guerrero, “Easy as Pi: A Network Coding Raspberry Pi Testbed” , Electronics , vol. 5, no. 67, 2016.

[7] P. Regis,,C. Miley and S. Sengupta, “ Multi-hop Mobile Wireless Mesh Network Testbed Development and Measurements”, International Journal of Innovative Research in Computer and Communication Engineering, vol. 5, no. 3, 2 017.

[8] Lakshman Naik.L, R.U.Khan, R.B.Mishra,"Analysis of Node Density and Pause Time Effects in MANET Routing Protocols using NS-3", International Journal of Computer Network and Information Security(IJCNIS), Vol.8, No.12, pp.9-17, 2016.

[9] Aditya Thapliyal, CRS Kumar,"Development of Data Acquisition Console and Web Server Using Raspberry Pi for Marine Platforms", International Journal of Information Technology and Computer Science(IJITCS), Vol.8, No.11, pp.46-53, 2016.

[10] Shelja Sharma, Suresh Kumar Bhardwaj,"Experimental Analysis of OLSR and DSDV Protocols on NS-2.35 in Mobile Ad-Hoc Networks", IJCNIS, vol.7, no.8, pp.21-29, 2015.

[11] Raspberry Pi Official website [Online], http://www.raspberrypi.org/ Accessed on 28 Jan 2018.

[12] Jacquet P., Muhlethaler P., Clausen T., Laouiti A, Qayyum A and Viennot L., ―Optimized link state routing protocol for ad hoc networks‖, In proceedings of the 5th IEEE Multi Topic Conference INMIC, pp. 62-68, 2001.

[13] V. Vesel´y, V. Rek, and O. Rysav´y, “Babel routing protocol for omnet++ more than just a new simulation module for INET framework,” CoRR, vol. abs/1609.05215, 2016.

[14] Clausen T. and Jacquet P., ―Optimized Link State Routing Protocol (OLSR)‖, RFC 3626, IETF Network Working Group, available online at https://tools.ietf.org/html/rfc3626, January 2018.

Authors’ Profiles

Salem Sati has a PhD in computer science from HHU University in Dusseldorf, Germany (2017), and an MSc degree in computer engineering from Higher Industrial Institute in Misurata, Libya (2008). Also he completed his BSc in computer engineering in Higher Industrial Institute in 1997. Dr. Sati has contributed to several local conferences in Libya, Also International IEEE conferences in North America. Asia and Europe in the fields of computer networks. He has many publications in IEEE conferences. He is currently a Doctor at the faculty of Information Technology in Misurata University.

Ahmed Elbareg is a lecturer at the department of Electronic Engineering in The Collage of Industrial Technology, Misurata. He had a BSc degree in computer Engineering from Higher Industrial Institute in Misurata (1994). and an MSc degree in computer engineering from Nottingham Trent University, England (2010). He has publication paper in International Electrical & Electronic Engineering and Technologies Conference in Turkey.

How to cite this paper: Salem Sati, Ahmed El-bareg," MANET Testbed using Raspberry PIs", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.8, No.2, pp. 52-63, 2018.DOI: 10.5815/ijwmt.2018.02.05