VOL. 10, NO. 3, FEBRUARY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

PERFORMANCE EVALUATION OF OLSR ROUTING PROTOCOL IN AD HOC NETWORK

Deni Lumbantoruan[1], Albert Sagala[2]
[1]Electrical Engineering Study Program, Institut Teknologi Del, Indonesia
[2]Jl. Sisingamangaraja Desa Sitoluama Laguboti, Indonesia
E-mail: toruan@del.ac.id

ABSTRACT

In this paper, we evaluated the performance of OLSR routing protocol in ad hoc network by using test bed method. Evaluation was conducted to know and understand the performance of ad hoc network in certain conditions and scenarios. Ad hoc network nodes were implemented by using Raspberry Pi with USB Wireless TP-Link WN722N. We have done three types of evaluation, such as: evaluate the availability of ad hoc wireless networks in various distance, evaluate the performace of multi hop ad hoc network, and the mechanism of self healing. Each testing is carried out on different circumstances and different scenarios. Based on some testing that already done in this research, it is known that ad hoc network with OLSR routing algorithm can overcome the problem of unreachable link between two nodes. Two nodes in one hop can reach ±180 metres with line of sight condition. Through testing, it is also shown that OLSR can handle the data delivery failures and find a new route to forward the data packets. Finally, to facilitate users in the future, we have remastered raspbian operating system with OLSR packet. It is hoped that users will be easier and faster to implement ad hoc network node with the Raspberry Pi devices.

Keywords: ad hoc, OLSR, Raspberry Pi, self healing, test bed.

INTRODUCTION

An Ad Hoc Network is a temporary network connection and does not depend on the existing of the infrastructure (C. S. R. Murthy, 2004). An ad hoc network also can be defined as a wireless multi hop network consists of collection of mobile nodes that are dynamic, spontaneous, and spread. Different with the centralized network infrastructure, in the ad hoc network each host is responsible for the delivery of data from sender to destination.

The utilization of ad hoc networks can be a solution when the network infrastructure cannot be used or infrastructure networks have limited services. For example, in the situation of natural disaster, when the infrastructure might be down or the capacity of the network overload so the temporary network as a replacement is needed. In addition, ad hoc network can also be applied for military purposes as shown in Figure-1 (Atacwireless, 2005). In the picture we can see that any vehicle or any person act as a node that can serve as hosts as well as routers.

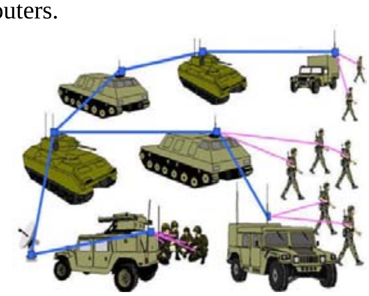

Figure-1. Ad hoc network for military network.

Ad hoc network has some characteristics such as: limited resources, frequent topology changes and the quality of the communication channel changing overtime. These makes a lot of paths between nodes become unstable and prone to error. Because of the characteristics of ad hoc network is very unique as mentioned above, ad hoc network topics still a hot topic to be discussed and researched until now.

Many researchers contributed in the research of ad hoc network which includes research in the physical layer, data link layer, network layer, especially in the fields of routing protocol, transport layer, and application layer. In each layers, every researchers tried to give suggestions to the new protocol, an update to the existing protocols, the comparison study between two or more protocols, and others.

In addition to many research which proposed new ideas about protocol or improvement of an existing protocol, there is also research area that is not less important in ad hoc network such as a research in evaluating the existing protocols either by simulation, mathematical analysis or test directly by using the real network devices which known as test bed. Specifically on test bed, research activities with test bed method aims to directly test the concept of ad hoc networks in a real network and to get the ad hoc network technologist closer commercial product development.

In this study, we evaluated the ad hoc network which uses OLSR routing algorithms. Ad hoc network devices are realized by using the Raspberry-Pi. After installation and configuration of ad hoc network device, then testing process is done through experiments on several real case scenarios in campus IT Del. Several scenarios were conducted such as the different number of hops scenarios, the different distance between nodes scenarios, and self-healing condition.

Furthermore, this paper is presented in several sections: Section 2 contains several studies related to our research. Section 3 contains the testing design. Section 4 contains the implementation, testing and analysis of testing results. Finally, the conclusions presented in section 5.

RELATED STUDY

As one of number of ad hoc network routing protocols, the optimized link state routing (OLSR) protocol (T. Clausen, 2003), was standardized by the IETF (RFC 3626). The OLSR protocol is a proactive routing protocol that uses an efficient link state packet forwarding mechanism. This protocol optimizes the pure link state routing protocol.

Optimizations are performed in two ways: by reducing the size of the control packets and by reducing the number of links that are used for forwarding the link state packets, i.e., flooding. Each node will select a set of neighboring nodes that will serve as a multipoint relay (MPR). The node selected as the MPR will be responsible for forwarding data packets to another node. The existence of the MPR allows an efficient mechanism for flooding during overhead by reducing the number of transmissions. Figure-2 gives an overview of the functions of the multipoint relay.

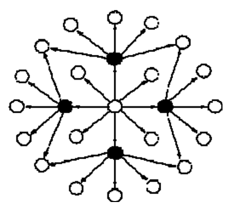

Figure-2. Multipoint relay.

Like other ad hoc network routing protocols, the performance of the OLSR routing protocol is strongly influenced by the available bandwidth on the radio link, and the network topology changes due to the random movement of nodes. This makes each node vulnerable to a less reliable connection so that packet loss increases. Increased packet loss results in increased retransmission processes. These retransmissions significantly increase end-to-end delay and reduce throughput because the communication overhead increases significantly.

Studies related to evaluation of ad hoc network using OLSR protocol and evaluation of ad hoc network with test bed method has been carried out by several groups of researchers. Studies by Makoto et al (Makoto Ikeda, 2008) perform performance evaluation of Optimized Link State Routing (OLSR) in Mobile Ad Hoc Network. In this studies, the performance of the OLSR in MANET in varies of data transferred was evaluated.

In study by Viviek et al (Vivek Srivastava 2008), experiments carried out by several inter-university teams. This experiment is an annual competition between universities in the world. The topic of the competition is Characterizing Mobile Ad hoc Network–The MANIAC Challenge Experiment, focus on Mobile Ad hoc Network Interoperability and Cooperation.

In research conducted by Megat Norulazmi et al (Megat N, 2013), community-based home security system using wireless mesh was developed. Prototype of community-based home security system is realized by using wireless mesh router and computer open mesh using Raspberry Pi. Studies by Ferdian (Ferdian Yunazar, 2012), Wireless Mesh Networks for Wireless Weather Station in Campus LIPI Bandung was implemented. Nana R. et al developed Wireless Mesh Nodes by using Soekris.net4801(Nana Rachmana, 2008). In this study, wireless mesh node hardware which is a core component of wireless mesh networks was developed using Soekris net4801. OLSRd used as a routing daemon.

In study conducted by Eric Erfanian (Eric Erfanian (2012), he outlined the steps to create a mesh network using the Raspberry Pi models. The device is built to support the mesh network that able to connect at speeds of 400kbps, where gateway is connected with 3 nodes.

In this paper, the implementation of an ad hoc network using Raspberry Pi is carried out. After implementation, then proceed with the testing of the network availability and network performance. Testing of ad hoc network done by using 5 nodes, where 3 nodes is realized by using the Raspberry Pi and 2 nodes by using a laptop. The parameters tested are network availability, condition of self-healing and communication delay for a wide variety of distance communication. Compared with the previous studies, this study is more real in terms of communication distance and scenarios.

PLANNING FOR TESTING

This section describes the preparations for implementing and testing of ad hoc networks with OLSR routing protocol, consist of hardware preperation, i.e. Raspberry Pi, and software preparation, i.e. operating system that will be installed on the device and OLSR package.

Testing preparation

Testing process starts by preparing hardware and software required for testing ad hoc networks. Software installation in hardware also became one of the preparations process. In this study, we use three pieces hardware Raspberry Pi with same specification. Those Raspberry Pi will serves as a node and as a router in routing process. Here is a list of the hardware needed to implement one node of ad hoc network.

ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

TABLE-1. list of hardware for one node.

Name of device	Description
Raspberry Pi board with B type (Broadcom BCM2835 SoC, processor 700 MHz, Low Power ARM1176JZ-F, 512MB SDRAM)	As a board that manages the entire activity, acts as a node in a mesh network.
Wireless adapter (TP Link WN722N)	As a signal transmitter and receiver so one node can interact with another node.
SD card (4 GB)	As a storage for software

Physical display of device Raspbian Pi model B is shown in Figure-3.

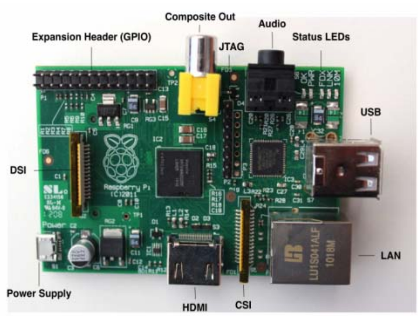

Figure-3. Device Raspberry Pi model B.

For design and implementation of ad hoc network devices using the Raspberry Pi need several softwares. The operating system used in this device is Raspbian, an operating system developed based on Debian. Raspbian used to optimize Raspberry Pi hardware. In addition, Raspbian is more than pure OS; this operating system developed with many additional software packages in a good format for making installation of Raspberry Pi device become easier.

For implementing and testing ad hoc networks, besides operating system, some software also needed such as OLSR routing packets used olsrd-0.6.6.1, Win32 Disk Imager needed for extracting operating system files to SD card before installation of operating system into the SD card.

Here is an ad hoc device installation phase:

1. Devices assembly

This phase is done by setting up Raspberry Pi boards and other supporting devices such as: wireless adapter, SD card, mouse, keyboard, monitor, and connecting cable.

2. Operating system installation

In this phase, raspbian operating system was installed on SD card.

3. OLSR package installation

In this phase, package olsrd-0.6.6.1 was installed on SD card.

4. Configuration

In this phase, the configuration of Raspberry Pi devices conducted such as OLSR package configuration and IP address configuration. If this configuration phase succeed, it will proceed to the stage of testing whether the device is functioning as a node for ad hoc networks.

Testing design

In this section we will describe some scenarios to test the ad hoc network. Testing consists of three parts, namely testing the range of ad-hoc wireless network, Self-Healing testing and testing of multi-hop condition.

Testing design for communication range

Topology used for testing the range of communication in ad-hoc wireless networks can be seen in Figure-4 below.

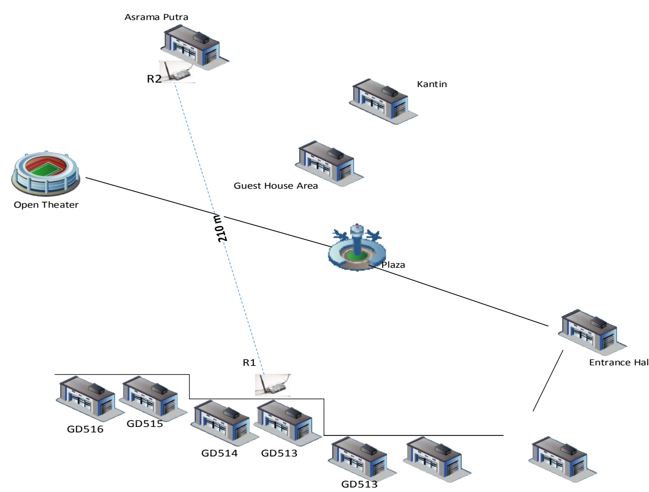

Figure-4. Topology for communication range testing.

In Figure-4, testing is done in the campus area of Del Institute of Technology. The addressing of devices is done as follows:

Device	IP address	Subnet mask
Raspberry Pi 1 (R1)	172.29.9.1	255.255.255.0
Raspberry Pi 2 (R2)	172.29.9.2	255.255.255.0

VOL. 10, NO. 3, FEBRUARY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

Testing design for self-healing

Topology used for testing Self-Healing in wireless ad hoc is shown in Figure-5. In this scenario, it was designed that connection between R3-R1 will switched to R3-R2 when R1 go down.

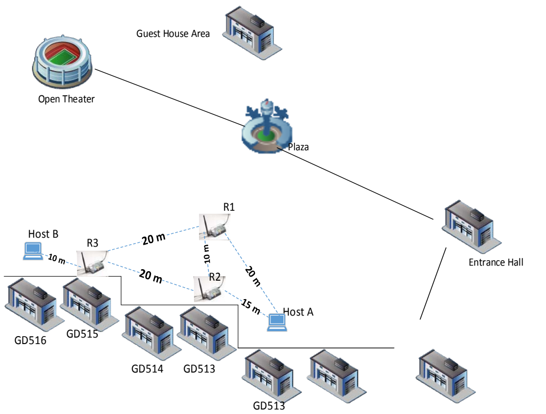

Figure-5. Topology for self-healing testing.

In the Figure-5, testing is done in the campus area of Del Institute of Technology. The addressing of devices is done as follows:

Device	IP address	Subnet mask
Raspberry Pi 1 (R1)	172.29.9.1	255.255.255.0
Raspberry Pi 2 (R2)	172.29.9.2	255.255.255.0
Raspberry Pi 3 (R3)	172.29.9.3	255.255.255.0
Host A	172.29.9.19	255.255.255.0
Host B	172.29.9.15	255.255.255.0

Testing design for multi-hop

Topology used for testing the multi hop condition of ad hoc wireless networks is shown in Figure 6. In this scenario we can see that ad hoc network consist of 5 hosts and 4 hops.

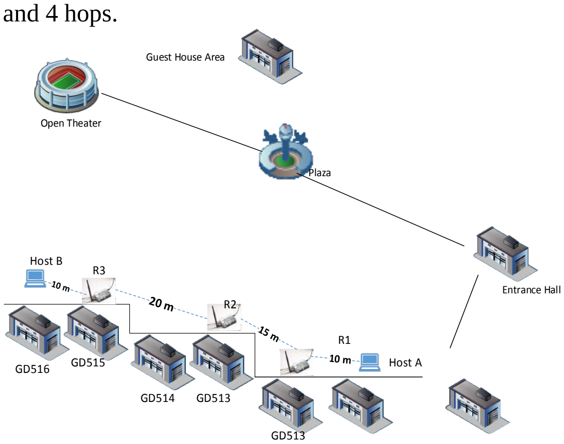

Figure-6. Topology for Multihop testing.

In the Figure-6, testing is done in the campus area of Del Institute of Technology. The addressing of devices is done as follows:

Device	IP address	Subnet mask
Raspberry Pi 1 (R1)	172.29.9.1	255.255.255.0
Raspberry Pi 2 (R2)	172.29.9.2	255.255.255.0
Raspberry Pi 3 (R3)	172.29.9.3	255.255.255.0
Host A	172.29.9.19	255.255.255.0
Host B	172.29.9.15	255.255.255.0

TESTING AND DISCUSSIONS

This section describes the test performed on the scenarios that have been designed in section 3.

Testing

Testing is carried out to ascertain whether ad hoc network that has been built by implementing the OLSR routing system able to carry out their task properly. Before testing the performance of OLSR routing protocol, it is needed to ensure the interconnection between nodes in ad hoc network.

Testing of hello message delivery

Hello message delivery is done to determine the nodes that are close together and connected each other. This message is broadcast to all nodes to find its neighbour and also has function to make changes in the routing table.

The delivery of hello message packet from one node to another node by using olsr routing protocol could be done with command: “olsrd –d 3”, so it will display the address of neighbor node as shown in Figure-7.

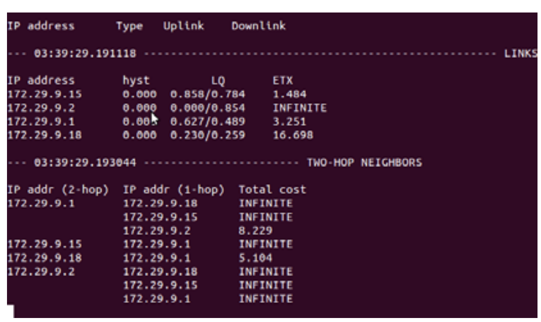

Figure-7. Hello message testing.

Network range testing

Testing the range of connections between nodes is done to evaluate the reach ability of connections between nodes. This is the first stage of testing before testing the OLSR routing implementation. This testing is done by sending ping from one node to another node with different distances.

The result of range possibility testing between two nodes in term of delays is shown in Figure-8.

ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

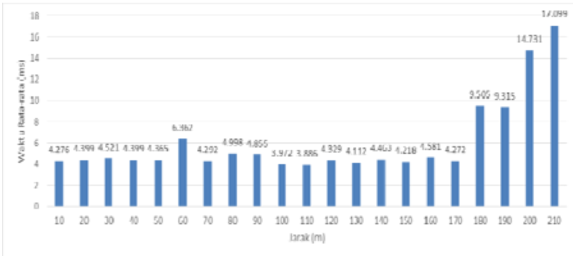

Figure-8. Delay on network range testing.

In each testing, with different distances, it was delivered the same number of packet, which is 20 packets. The graph in Figure-8 shows that there was no significant difference in the performance of the connection until at a distance 170 meters. Packet loss in each testing process could also be seen in Figure-9.

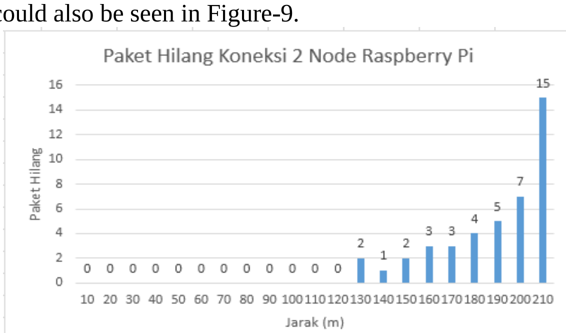

Figure-9. Packet loss on network range testing.

In the graph above can be seen that up to distance 120 meters, there is no packets lost yet. However, from a distance 130 to 210, there are some packets lost. The tolerance of packets lost that can be received is 20%. So, it can be concluded that the wireless network is still works optimal up to a distance of 180 meters. Because at a distance of 180 meters, from 20 packets there are 4 (20%) packet loss, and 16 (80%) packet received.

Self-healing testing

Self-healing testing in OLSR is done to ascertain whether the node in OLSR ad hoc network can still send packets to the destination through other channels if one pathway is broken or the neighbor terminal is down. This self-healing mechanism aims to find a new route to send packet to the destination. Several steps to perform OLSR self-healing as shown in Figure-5:

- Host B ping host A. By using trace route command, it can be seen the route to get to host A, as follows:

```
root@fun-Lenovo-8460:/home/fun:# traceroute 172.29.9.19
traceroute to 172.29.9.19 (172.29.9.19), 30 hops max, 60 byte packets
1 172.29.9.3 (172.29.9.3) 4.488 ms 5.169 ms 5.152 ms
2 172.29.9.1 (172.29.9.1) 31.594 ms 38.823 ms 38.814 ms
3 172.29.9.19 (172.29.9.19) 35.594 ms 48.823 ms 38.814 ms
```

- At the time of ping process running, node R1 was turned off. Event R1 was down, ping process is still running. We find that ping process can be interrupted for about 1 minute because OLSR has not update their routing tables yet, and it automatically connect again after the routing table is updated.
- By using trace route command from host B to host A again, it was found that the routing has changed from R2 go to R3 and then to host A as follows.

```
root@fun-Lenovo-8460:/home/fun:# traceroute 172.29.9.19
traceroute to 172.29.9.19 (172.29.9.19), 30 hops max, 60 byte packets
1 172.29.9.3 (172.29.9.3) 3.423 ms 5.169 ms 4.152 ms
2 172.29.9.2 (172.29.9.2) 39.594 ms 38.823 ms 38.814 ms
3 172.29.9.19 (172.29.9.19) 60.594 ms 68.823 ms 48.814 ms
```

Self-healing testing with application

In the previous testing, self healing testing done by using ping methode. In this section, testing was performed by acsessing the web server connected to ad hoc network as shown in Figure-5.

Here are several tests to perform OLSR self-healing using application:

1) One of the host acts as a web server and another host acts as a client to access the web server. Raspberry Pi devices serve as an intermediate node. When the connection from R3 to R1 being disconnected while the host is requesting process to the web server, then the process will stopped temporarily. However, after refreshing process, then the web application on the server can be accessed again. This occurs because OLSR as routing protocols have found the new route from the host to the web server.
2) One of the host acts as a chatting server by installing openfire and other hosts acts as a client which uses a spark application for chatting. When doing chat and routes between hosts with the server is lost due to one of the intermediate node is down, then chat application will be interrupted for about 1 minute and it will be re-activated automatically.
3) One host acts as a video server and another host acts as a client to access the video through the browser. Video playback will be interrupted if the routing is lost, and will reconnect automatically when a new routing has been discovered and video playback should work again.

OLSR multi-hop testing

OLSR Multi-hop testing is done to ensure that a node can send packets to the destination through other nodes as intermediate node. Testing is done based on test scenarios as shown in Figure-6.

Before conducting the test, it must be confirmed that:

- Testing is done in real condition, not use a routing manipulation such as iptables.

- Host A cannot ping directly to R3 and Host B
- Host B cannot ping directly to the R1 and Host A

Testing stage was performed as:

1. Host A send ping to the adjacent node R1 with 1 hop connection. Trace route command done to see the route from node A to node R1. The results showed that host A directly connected to node R1.

```
root@laban-Lenovo-8460:/home/laban:#traceroute 172.29.9.1
traceroute to 172.29.9.1 (172.29.9.1), 30 hops max, 60 byte packets
1 172.29.9.1 (172.29.9.1) 7.713 ms 7.674 ms 8.814 ms
```

2. Host A send ping to R2 which has 2 hop connection. The results showed that host A is connected to node R2 through node R1 as shown below.

```
root@laban-Lenovo-8460:/home/laban:#traceroute 172.29.9.2
traceroute to 172.29.9.2 (172.29.9.2), 30 hops max, 60 byte packets
1 172.29.9.1 (172.29.9.1) 4.488 ms 5.169 ms 5.152 ms
2 172.29.9.2 (172.29.9.2) 31.594 ms 38.823 ms 38.814 ms
```

3. Host A send ping to R3 through intermediate node R1 and R2. The results showed that host A is connected to node R1, R2, and then to R3.

```
root@laban-Lenovo-8460:/home/laban:# traceroute 172.29.9.3
traceroute to 172.29.9.2 (172.29.9.2), 30 hops max, 60 byte packets
1 172.29.9.1 (172.29.9.1) 3.152 ms 3.110 ms 5.545 ms
2 172.29.9.2 (172.29.9.2) 11.351 ms * 44.515 ms
3 172.29.9.3 (172.29.9.3) 91.623 ms 91.633 ms 91.617 ms
```

4. Host A send ping to host B through intermediate node R1, R2 and R3. The results as shown below:

```
root@laban-Lenovo-8460:/home/laban:#traceroute 172.29.9.15
traceroute to 172.29.9.15 (172.29.9.15), 30 hops max, 60 byte packets
1 172.29.9.2 (172.29.9.2) 16.619 ms 16.546 ms 27.215 ms
2 * * 172.29.9.3 (172.29.9.3) 193.727 ms
2 172.29.9.15 (172.29.9.15) 303.213 ms 307.006 ms 311.424 ms
```

DISCUSSIONS

Through intensive testing process has confirmed that ad hoc network which implemented by using Raspberry Pi with OLSR routing protocol can be used properly for multi-hop communication function or in communication with self-healing condition.

From testing that has been done can be seen that the self-healing capability to handle the network availability is very effective because there is no significant difference in term of connection speed between self-healing multi-route and a direct connection. In the experiment can be observed that the application browsing, chatting and video playback from the server is still running well when the intermediate nodes go down. The delay perceived relatively small and the communications can be resumed quickly. This situation is possible because in the OLSR routing protocol, every node send Hello Message periodically to update their routing table and maintain the connection. When there is any intermediate node broken, then the neighbor nodes will try to find another route quickly based on the up to date routing tables. This mechanism enables self-healing process running well, where user does not aware there is a problem in the intermediate node.

Through testing has also confirmed that the multi-hop communication can be realized in ad hoc network. Through experiments it is known that the delay will increase when the number of hops increases. This delay comes from the accumulation of process delay, transmission delay and propagation delay. From the delay we can see that data communication which transmitting small size data, such as data text, can still be forwarded well on 4 hops communication.

To determine path taken to a node, trace route sends three probe packets with UDP types from different source ports, with TTL values 1. When the packet reaches the next-hop router, the TTL of the packets will be decrease so that it becomes 0, and the next-hop router will reject the UDP packet while sending ICMP Time-to-Live Exceeded trace route to the origin node. By this way, the sender knows the IP address of the first path taken.

Then, the source of trace route sends again 3 pieces of UDP packet with TTL value is increased 1 become 2, so the first router in the path toward the destination of trace route will skip the UDP packet to the next router. The second hop router will see that the package has expired, TTL become 1 after deduction by the first router. So same as the first router, that router will send the ICMP packet Time-to-Live Exceeded the source of trace route.

As the source of traceroute already known the seccond hop in path to destination, source of the traceroute will send again 3 pieces of UDP packet with TTL value is increased 1 become 3.

The third hop router will respond with ICMP packet Time-to-Live Exceeded to the source of trace route, so the source of traceroute knows the IP address of third hop router. This process will be repeated for sending UDP packet to destination of trace route. Three packets of UDP trace route is the default packet from trace route applications.

Each data packet has time to live (TTL). Time to live will decrease by 1 each data packet forwarding through one node. OLSR as a routing protocol will determine route which traversed by packet. More hops are traversed by a packet to destination, then packets will require more time in transmission. In other words, packet delay will increase as number of hops increase.

ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

Remastering Raspbian image

Re-mastering of raspbian image is a process to duplicate raspbian operating system and saved as an image file. Re-mastering of raspbian image is needed to help researcher or developers who want to explore further about the Raspberry Pi and implement ad hoc networks using Raspberry Pi and OLSR as routing protocol. By using raspbian image, time for learning and implementing OLSR in Raspberry Pi will decrease. It is not neccessary anymore to do the same way of configuration because all of configuration and services features needed already installed and configured properly.

CONCLUSIONS

Based on testing result includes testing the availability of ad hoc networks with 3 and 4 hops, testing ad hoc network coverage, testing ad hoc network routing, it can be concluded that the implementation of ad hoc networks with OLSR routing by using Raspberry Pi devices as nodes can be implemented. Through several testing scenarios can be seen that the ad hoc network built can be used to overcome the limitations within range, by implementing of multi hop communication.

By doing testing with scenarios such as range testing, multi-hop testing, and wireless adhoc network self-healing can be concluded that the Raspberry Pi by using USB Wireless TP-Link WN722N with OLSR routing protocol can work well with distances up to ± 180 meters between two nodes in one hop in line of sight communication. When the number of hops increase then it will decrease delivery ratio and increase the delay, but the system can still work well. Through testing we can also find that the self-healing mechanism performed successfully. Testing by using video access applications on the server shows that eventhough the intermediate node switch to another node, users can still access the video server properly.

REFERENCES

C. S. R. Murthy (2004), “Ad hoc Wireless Network, Architecture and Protocols”, Prentice Hall.

Atacwireless. (2005) http://www.atacwireless.com/adhoc.html

T. Clausen. (2003), RFC 3626, Optimized Link State Routing Protocol (OLSR).

Makoto Ikeda. (2008), Experimental and Simulation Evaluation of OLSR Protocol for Mobile Ad-Hoc Networks, Proceedings on 2nd International Conference, NBiS 2008, Turin, Italy, September 1-5, 2008.

Vivek Srivastava. (2008), Characterizing mobile ad hoc networks: the maniac challenge experiment, Conference: Proceedings of the Third ACM Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization, WINTECH 2008, San Francisco, California, USA.

Megat N. (2013), Community based home security system using wireless mesh network, Malaysia: Department of MIIT, University of Kuala Lumpur.

Ferdian Yunazar. (2012), Wireless Mesh Network Implementation for Data Communication Network in Wireless Weather Station, Pusat Penelitian Informatika Lembaga Ilmu Pengetahuan Indonesia.

Nana Rachmana. (2008), Wireless Mesh Node Development with Soekris Net4801, Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung.

Eric Erfanian. (2012), Mobile Mesh Network with the Raspberry Pi, http://www.ericerfanian.com/mobile-mesh-network s-with-the-raspberry-pi-part-1.