

Enhancing the 'Willingness' on the OLSR Protocol to Optimize the Usage of Power Battery Power Sources Left

Saaidal Razalli Azzuhri

*Electrical Engineering Department, Faculty of Engineering
University of Malaya
Kuala Lumpur, 50603 Federal Territory, Malaysia*

saaidal@um.edu.my

Suhazlan Suhaimi

*Faculty of Information & Communication Technology
Sultan Idris Education University
Tanjong Malim, 35900 Perak, Malaysia*

suhazlan@ftmk.upsi.edu.my

K. Daniel Wong

*Department of Information Technology
Malaysia University of Science & Technology
Petaling Jaya, 47301 Selangor, Malaysia*

dwong@must.edu.my

Abstract

Mobile ad-hoc networks are infrastructure-free and highly dynamics wireless networks. There are many routing protocols for Mobile Ad Hoc Networks (MANET). One of the popular routing protocols is Optimized Link State Routing (OLSR). OLSR was developed to work independently from higher-layer protocols and it creates an underlying architecture for communication without the help of traditional fixed-position routers. In addition, OLSR attempts to maintain the communication routes for the mobile nodes that have limited transmission range. This paper discusses the OLSR implementation in terms of battery power status and its advantages particularly pertaining to the relationship between the power status function with the OLSR by modify some of the OLSR source code. The study also focuses on maximizing the use of the battery power sources. The results from the experiment show that the usage of the battery power sources left was maximally used when the "Willingness" for the nodes was indeed increased. We will conclude that the experiments have been successfully done and the results demonstrate the improvement for the OLSR nodes to maximize the battery power sources usage on the MANET by enhancing the "Willingness" on the OLSR protocol.

+

Keywords: Multipoint-relay, ad-hoc, cross platform, OLSR, MANET

1. INTRODUCTION

A Mobile Ad Hoc Networks (MANET) is an autonomous system of mobile nodes. It consists of mobile platforms for example a router with multiple hosts and wireless communications devices. Herein simply referred to as 'nodes' which are free to move about arbitrarily. It also may operate in isolation or may have gateways to and interface with fixed network. There are many important

research questions in MANET. However, power efficiency is one of the most important issues. It is important to realize that issues such as QoS support, TCP performance, speed of routing repair process and others are secondary if nodes have a high probability of running out of energy resources. Energy awareness in wireless ad hoc networks actually spans across several communication layers. Advances in battery technology are very slow compared to the results achieved in integrated circuit technology particularly in comparison to the rate of growth in communication speeds. Therefore, saving transmission power represents one of the most significant methods for long term wireless system performance.

In OLSR, the power energy resource is a very important factor in deciding whether the node can forward a packet or not. It is because only nodes where the "Willingness" is bigger than 7 can be used to forward packets. The higher the willingness of the node the higher the chance it will be selected as an MPR (we will explain the terms "willingness" and MPR later). The nodes willingness is based on the power resources it has from AC power sources or from the battery. There is no power problem when the nodes are using the power from the AC power sources as the nodes will get power consistently. However, problems may occur when the nodes use the power from the battery in which the power is limited. On the other hand, the MANET nodes are usually used in scenarios without any infrastructure, and therefore without AC power sources [1]. Thus, it is important to optimize the usage of the limited resources from the battery.

2. BACKGROUND THEORY

2.1. OPTIMIZED LINK STATE ROUTING (OLSR)

The Optimized Link State Routing Protocol (OLSR) is a protocol that was developed for mobile ad hoc network (MANET). It is a variation of traditional link state routing, modified for improved operation in ad hoc networks. It is a table driven and proactive routing protocol where the nodes exchange their topology information with other nodes regularly. The routes in the proactive protocols are always immediately available when needed [2]. OLSR is designed to work in a completely distributed manner and does not depend on any central entity. The protocol does not require reliable transmission of control messages. Each node sends control messages periodically and sustains a reasonable loss of some such messages. Such losses occur frequently in radio networks due to collisions or other transmission problems.

The key feature of this protocol is multipoint relays (MPRs). Topological changes cause floods of the topological information to all available nodes in the network. Therefore, the multipoint relays (MPRs) are used to reduce the overhead of network floods and size of link state updates. Every node selects a set of its neighbour nodes as multipoint relays (MPRs). Only MPRs nodes are responsible for forwarding control traffic. MPRs nodes only declare link-state information when the requirements for OLSR provide the shortest path information message [3]. The neighbours, whom the node selects as MPR, announce this information periodically in their 'Control Message'. In route calculation, the MPRs are used to form the route from a given node to any destination in the network. It is also be used to facilitate efficient flooding of Control Message in the network. OLSR uses two kinds of Control Messages; Hello Messages and Topology Control messages.

2.1.1.HELLO MESSAGE

Each node should detect the neighbour nodes with which it has a direct and symmetric link. The uncertainties over radio propagation may make some links asymmetric. Consequently, all links MUST be checked in both directions in order to be considered valid. To accomplish this, each node broadcasts HELLO messages, containing information about neighbours and their link status. The link status may either be "symmetric", "heard" (asymmetric) or "MPR". "Symmetric" indicates that the link has been verified to be bi-directional, for example, it is possible to transmit data in both directions. "Heard" indicates that the node can hear HELLO messages from a neighbour, but it is not confirmed that this neighbour is also able to receive messages from the node. "MPR" indicates that a node is selected by the sender as a MPR. A status of MPR further implies that the link is symmetric. These control messages are broadcast to all one-hop neighbours, but are *not relayed* to further nodes. A HELLO-message contains [4]:

- A list of addresses of neighbours, to which there exists a symmetric link.

- A list of addresses of neighbours, which have been "heard".
- A list of neighbours, which have been selected as MPRs.

The list of neighbours in a HELLO message can be partial (for example, due to message size limitations, imposed by the network), the rule being that all neighbour nodes are cited at least once within a predetermined refreshing period (HELLO_INTERVAL).

2.1.2. TOPOLOGY CONTROL (TC)

In order to build the topology information base needed, each node, which has been selected as MPR, broadcasts Topology Control (TC) messages. TC messages are flooded to all nodes in the network and take advantage of MPRs. MPRs enable a better scalability in the distribution of topology information. A TC message is sent by a node in the network to declare its MPR Selector set. For example, the TC message contains the list of neighbours which have selected the sender node as a MPR. The information diffused in the network by these TC messages will help each node to calculate its routing table. A node which has an empty MPR selector set, such as nobody has selected it as a MPR, MUST NOT generate any TC message.

2.1.3. CONTROL TRAFFIC

All OLSR control traffic is transmitted over UDP on port 698. The Internet Assigned Numbers Authority (IANA) assigns this port to OLSR. The RFC states that this traffic is to be broadcasted when using IPv4 however, no broadcast address is specified. It is implicitly understood that one must use a multicast address when using IPv6 as IPv6 does not have broadcast addresses, even though it is not specified in the RFC [5]. FIGURES 2.1 and 2.2 shows the paths information is passed when being spread, first using regular flooding, then using MPR flooding. The number of retransmissions in a MPR scenario highly depends on the network topology and the MPR calculation algorithm. However, using the same topology as in FIGURE 2.3, a possible MPR calculation could lead to the yellow nodes in FIGURE 2.4 being chosen as MPRs by the center node. As one can see, if the center node is to flood a message throughout the network, four retransmissions are done using MPR as opposed to 24 using traditional flooding.

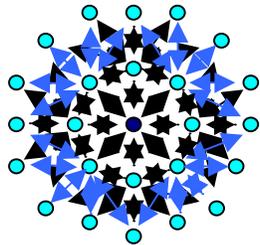


FIGURE 2. 1: Flooding a packet in a wireless multihop network. The arrows show all transmission (T. Andreas, 2004).

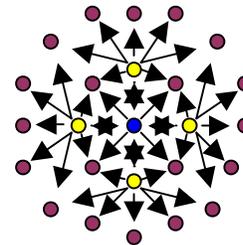


FIGURE 2. 2: Flooding a packet in a wireless multihop network from a centre node using MPRs (Yellow). The arrows show all transmission (T. Andreas, 2004)

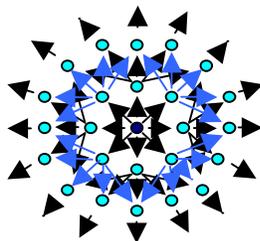


FIGURE 2. 3: Flooding a packet in a wireless multihop network. The arrows show the way information is passed but not to all transmission (T. Andreas, 2004).

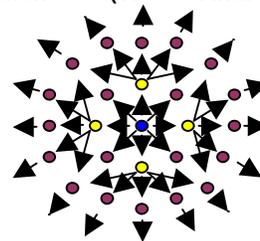


FIGURE 2. 4: Flooding a packet in a wireless multihop network from a centre node using MPRs (Yellow). The arrows show the way information is passed but not to all transmission (T. Andreas, 2004).

2.1.4. ROUTING

LINK SENSING

Link Sensing is accomplished through periodic emission of HELLO messages over the interfaces through which connectivity is checked. A separate HELLO message is generated for each interface and emitted in correspondence with the provisions. Resulting from Link Sensing is a local link set which describing links between "local interfaces" and "remote interfaces" for example the interfaces on neighbour nodes. If sufficient information is provided by the link-layer, this may be utilized to populate the local link set instead of HELLO message exchange. Link sensing populates the local link information base. Link sensing is exclusively concerned with OLSR interface addresses and the ability to exchange packets between such OLSR interfaces. The mechanism for link sensing is the periodic exchange of HELLO messages.

The Link Set is populated with information on links to neighbour nodes. The process of populating this set is denoted "link sensing" and is performed using HELLO message exchange, updating a local link information base in each node. Each node should detect the links between itself and neighbour nodes. Uncertainties over radio propagation may make some links unidirectional. Consequently, all links MUST be checked in both directions in order to be considered valid. A "link" is described by a pair of interfaces: a local and a remote interface. For the purpose of link sensing, each neighbour node (more specifically, the link to each neighbour) has an associated status of either "symmetric" or "asymmetric". "Symmetric" indicates that the link to that neighbour node has been verified to be bi-directional, for example, it is possible to transmit data in both directions. "Asymmetric" indicates that HELLO messages from the node have been heard (for example, communication from the neighbour node is possible), however it is not confirmed that this node is also able to receive messages (for example, communication to the neighbour node is not confirmed). The information, acquired through and used by the link sensing, is accumulated in the link set. (A. Laoutti, P. Muhlethaler, A. Najid and E. Plakoo, 2002). The process will illustrate as a FIGURE 2.5.



FIGURE 2.5: A typical neighbour discovery session using HELLO message

MULTIPOINT RELAYS

The Multipoint Relays (MPR) is the key idea behind the OLSR protocol in order to reduce the information exchange overhead. In OLSR only the MPRs can forward the data throughout the network. Each node must have the information about the symmetric one hop and two hop neighbours in order to calculate the optimal MPR set. The two hop neighbours are found from the Hello message because each Hello message contains all the nodes' neighbours [7]. FIGURE 2.6 shows the multipoint relays (MPRs) selection between its neighbours. For instance node A select m1, m2 and m3 nodes. When node A floods a message, only node m1, node m2 and node m3 re-transmit it, after that the MPRs of node m1, node m2 and node m3 retransmit and so on. There are 2 rules to select a node for MPR:

- First, any 2-hop neighbour must be covered by at least one multipoint relay.
- Second, try to minimize the number of multipoint relays.

Then a node forwards a flooding packet according to several rules. The multipoint relays idea is to reduce the redundant retransmissions in the same region by minimizing the overhead of flooding message in the network. Each node in the network selects a set of nodes in its symmetric 1-hop neighbourhood, which may retransmit its messages. This set of selected neighbour nodes is called the "Multipoint Relay" (MPR) set of that node. The neighbours of node N which are *NOT* in its MPR set, receive and process broadcast messages but do not retransmit broadcast messages received from node N.

Each node selects its MPR set from among its 1-hop symmetric neighbours. This set is selected such that it covers (in terms of radio range) all symmetric strict 2-hop nodes. The MPR set of N, denoted as $MPR(N)$, is then an arbitrary subset of the symmetric 1-hop neighbourhood of N, which satisfies the following condition: every node in the symmetric strict 2-hop neighbourhood of N must have a symmetric link towards $MPR(N)$ (FIGURE 2.7). The smaller a MPR set, the less control traffic overhead results from the routing protocol gives an analysis and example of MPR selection algorithms.

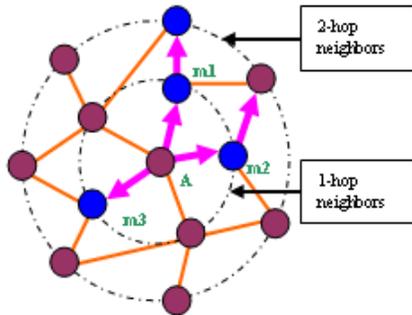


FIGURE 2.6: The Selection of MPRs among its neighbours.

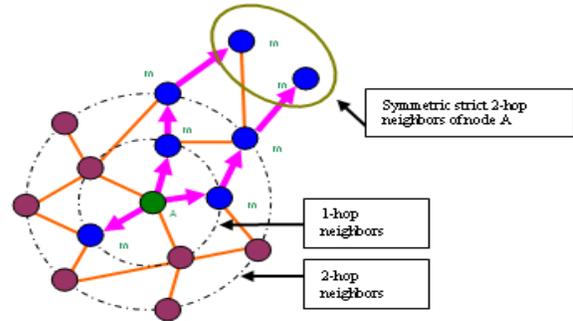


FIGURE 2.6: Symmetric strict 2-hop neighbours of node A

Each node maintains information about the set of neighbours that have selected it as MPR. This set is called the "Multipoint Relay Selector set" (MPR selector set) of a node. A node obtains this information from periodic HELLO messages received from the neighbours. A broadcast message, intended to be diffused in the whole network, coming from any of the MPR selectors of node N is assumed to be retransmitted by node N, if N has not received it yet. This set can change many time (for example, when a node selects another MPR-set) and is indicated by the selector nodes in their HELLO messages. Another rule that apply here is re-emitting rule that is shown in FIGURE 2.8. In FIGURE 2.8, if node m5 receives from node m2 first, it will re-transmit but, if the node m5 receives from node m1, it will not re-transmit. It means that no node will be missed when OLSR is used.

SHORTEST PATH WITH MPR LINKS

The MPR links offer a sparse partial topology containing the shortest paths. Any 2-hop neighbour of n node of a source of A node must have selected by some neighbours of A node as a MPR since an A node is a 2-hop neighbour of n node. Indeed, any node at distance m6 node from A node must have selected as MPR. Thus, it used MPR links backward to route from A node to m6 node.

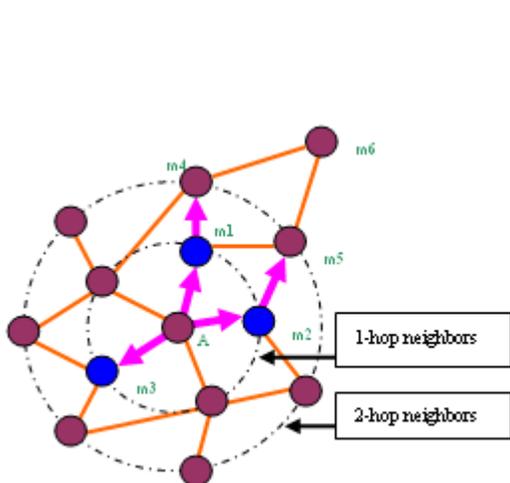


FIGURE 2.7: Re-emitting rule

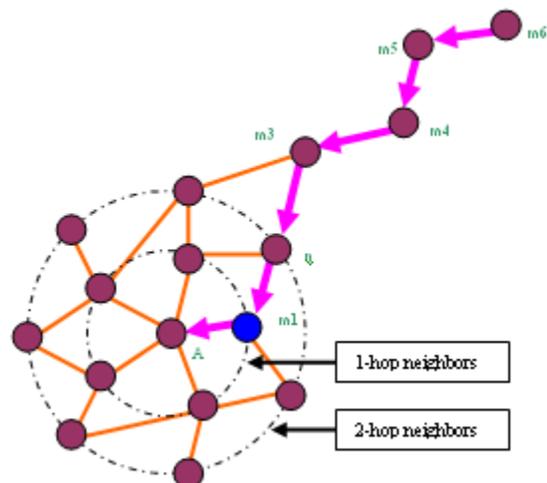


FIGURE 2.8: The shortest path with MPR link

TOPOLOGY ADVERTISEMENT IN OLSR

The MPR links are flooded in every node. Every node is able to compute the shortest path routes to every destination. On the other hand classical link state protocols will flood entire neighbourhoods, consuming additional bandwidth for just getting redundant routes.

A) MULTIPOINT RELAYS SELECTION

The objective of MPR selection is for a node to select a subset of its neighbours such that a broadcast message, retransmitted by these selected neighbours, will be received by all nodes 2 hops away. The MPR set of a node is computed such that it, for each interface, satisfies this condition. The information required to perform this calculation is acquired through the periodic exchange of HELLO messages, as described in section 6 in the RFC 3626. The MPR Selections algorithm that propose by RFC 3626 constructs the MPR set which includes minimum number of the one hop symmetric neighbours from which it is possible to reach all the symmetrical strict two hop neighbours. The node must have the information about one and two hop symmetric neighbours in order to start the needed calculation for the MPR set. All the exchange of information is broadcasted using Hello messages (H. Aleksandr, 2004).

B) TOPOLOGY SELECTION

The nodes that are selected as MPR need to send the topology control (TC) message in order to exchange and build topological information base. Only MPRs are allowed to forward TC messages, in which TC messages are broadcasted throughout the network. In order to advertise its own links, TC message must be sent by a node in the network. The node must send at least the links of its MPR selector set. The TC message includes the own set of advertised links and the sequence number of each message. The sequence number is used to avoid loops of the message, if the node gets a message with the smaller sequence number, it must discard the message without any updates. The node must increment the sequence number when the links are removed and added from the TC message (H. Aleksandr, 2004). When the nodes advertised, links set becomes empty and it can still send an empty TC messages for specified amount of time, in order to invalidate previous TC messages. The size of the TC message can be quite big, so the TC message can be sent in parts during some specified amount of time. Node can increase its transmission rate to become more sensible to the possible link failure.

C) ROUTING TABLE CALCULATION

The node maintains the routing table in which the routing table entries have "destination address", "next address", "number of hops" to the destination address and "local interface address". "Next address" indicates the next hop node. The information can be get from the topological set (TC messages) and the local link information base (Hello messages). Therefore, any changes for the set, the routing table will be recalculated (P. Jacquet, A. Laouitti, P. Minet and C. Viennot, 2001).

2.2. MULTIPOINT RELAY (MPR COMPUTATION)

The detail description that specifies a procedure of the proposed heuristic and the terminology that used to describing the heuristic will get in RFC 3626 document for selection of MPRs. It constructs a MPR-set, which enables a node to reach any node in the symmetrical strict 2-hop neighbourhood through relaying on one MPR node with willingness that is different from WILL_NEVER. The heuristic MUST be applied per interface. The MPR set for a node is the union of the MPR sets found for each interface. The implementation of the algorithm is located in /src/mpr.c which is the mpr.c source code file was automatically installed when installing the OLSR software. All one-hop neighbours are linked to the two-hop neighbours that can be reached. Therefore, when selecting a neighbour as MPR, all corresponding two-hop neighbours can easily be updated to reflect this. The algorithm removes all previous selected MPRs and recalculates the whole MPR set. This is done by traversing the neighbour set based on the registered willingness of neighbours, starting with a willingness of 7 and decreasing down to 1. Nodes with a willingness of 0(WILL_NEVER) are never selected as MPRs while nodes announcing a willingness of 7(WILL_ALWAYS) will always be selected as MPRs.

2.3. WILLINGNESS (SETTING WILLINGNESS)

“Willingness” is a willing or interest of the node in the Ad Hoc network to give a contribution or commitment to the other nodes in order to send a data in the network. In OLSR, willingness is set based on the power-status of the node. This information is extracted from the pseudofile /proc/apm, which is the user-space interface to the *Advanced Power Management* offered by the kernel. Advanced Power Management (APM) is the predecessor to ACPI. The BIOS needs to handle all power management which devices are being set into lower power status based on the device activity timeouts. In this case, if no such file is present, willingness will be set to WILL_DEFAULT (3). The user can also set a fixed value for willingness in the configuration file. The willingness is based on a trivial calculation. Beneath is the snippet of code that calculates willingness. This code is implemented in the function “olsr_calculate_willingness” in src/olsr.c. This olsr.c source code file was installed automatically into the nodes computer when doing the installation of the OLSR software.

```

/* If AC powered */
if(ainfo.ac_line_status)
    return 6;

/* If battery powered
 *
 * juice > 78% will: 3
 * 78% > juice > 26% will: 2
 * 26% > juice will: 1
 */
return (ainfo.battery_percentage / 26);
    
```

FIGURE 2.9: Source code for “olsr_calculate_willingness” function in olsr.c source code file

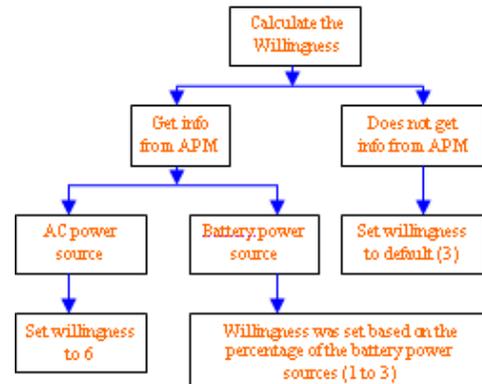


FIGURE 2.10: The willingness calculation flow diagram

2.4. OLSRD PLUG-IN

In order to maintain the communication session, process and generate packets, OLSR allows the usage of dynamically linked plug-ins, which is able to access the necessary functionality in it. Plug-ins can be used for almost everything because the plug-in interface offers a multi-purpose call to perform its specialized tasks. The OLSR daemon (olsrd) is support dynamic loading of plug-in (dynamic link library) for generation and processing of private package type starting from the version 0.4.3. The OLSR version that we used on the experiment was version 0.4.10. The design has been chosen because it does not need to change any code in olsrd in order to add custom package of functionality. The plug-in also can be written with any languages that can be compiled as a dynamic library. Thus, a person does not need to use the extended OLSR functioning to rely on heavy patching in order to maintain functionality whenever the new olsrd version is released.

2.5. THE POWERSTATUS PLUG-IN

The powerstatus plug-in provides a function where the node periodically floods the network with a custom packet, containing the following information:

- Whether or not the node is battery powered
- Estimated usage time left on the battery if using a battery
- Percentage of power left on the battery if battery powered.

It is very useful to inform the other nodes about the status of the power info itself. This information is very useful for the node to calculate the willingness for the usage of the other node to select a MPR. The details about the MPR and willingness have been discussed earlier of this chapter.

3.0 METHODOLOGY

3.1. RESEARCH TOOLS

We use specific tools for doing the experiments such the operating system, hardware, software

and also the environment of the testbed. We need to verify the specific tools because the different tools that we used sometimes give some effect to the result that we collect. It is also important because we need to make sure that all the element that we used for the tools are applicable and can work well without any error.

3.1.1. OPERATING SYSTEM REQUIREMENTS

The platform used Linux operating system (Fedora Core 5 and 6) in the experiment because the updated development of OLSR was created on it. In order to make sure the OLSR software could work with the latest Linux operating system, the latest version of Linux was chosen for experiment ensuring the required capabilities. Thus, the latest Linux kernel version (version 2.6.19) was used in the experiments ensuring that the results from the experiments are reliable.

3.1.2. HARDWARE REQUIREMENTS

In order to achieve greater mobility, wireless notebook devices were used to perform the OLSR functions. Other devices such as PDAs or hand phones are also suitable with OLSR depending on the operating system that supports the hardware. For this testbed, three DELL's notebooks Latitude D500 named as Telco1, Telco2 and Telco3 are been used. Another notebook that has been used in the experiment was Acer Aspire 5580, which was given the name Acer. The detail of the specifications and configurations is shown in TABLE 3.1.

	Dell Latitude D500	Acer Aspire 5580
Processor	Intel Pentium M Processor 100 MHz	Intel Core 2 Duo Processor 1.66 GHz
Memory	256 MB RAM	512 MB RAM
Storage	40 GB HDD	80 GB HDD
Network Adapter	<ul style="list-style-type: none"> Intel Pro/100 VE Network Connection Intel Pro/Wireless LAN 2100 3A Mini PCI Adapter 	<ul style="list-style-type: none"> Intel PRO/Wireless 3945ABG Network Connection Marvell Yukon 88E8038 PCI-E Fast Ethernet Controller
Operating System	<ul style="list-style-type: none"> Windows XP Pro Fedora Core 5/6 	<ul style="list-style-type: none"> Windows XP Home Fedora Core 6
Other Device	<ul style="list-style-type: none"> CDRW Drive 	<ul style="list-style-type: none"> COMBO Drive Bluetooth

TABLE 3. 1: The hardware specification and configuration for OLSR nodes.

3.1.3. SOFTWARE REQUIREMENTS

The experiments used the latest version of the OLSR in the implementation of the testbed utilizing the olsrd-0.4.10 version. There are a lot of enhancements in OLSR, entailing the use of the latest version that is free from bugs compared to the previous version. Therefore, the latest version is better and more reliable for use in the experiment owing to its latest features of OLSR software. The 'gcc' (gnu c compiler) is one of the prerequisite software for the OLSR that should be installed in the Linux for OLSR software compilation. Others software or tool the used was wireshark software, ping and traceroute or tracet tools.

3.1.4. TESTBED REQUIREMENT

This experiment was tested on MUST campus environment, which involved the first floor and second floor of wing C comprising the Wireless room, IT lab, library and the corridor in the first and second floor for testbed. The results would differ if the experimental requirements chosen were altered.

3.2. OLSR SOURCE CODE MODIFICATION

The focus of the experiment is on maximizing the battery usage and the OLSR performance namely the modification of the core and the power status plug-in on the OLSR software to determine the change of willingness that relate to the usage of power resources based on the AC power or battery power sources. At this juncture, the willingness can be enhanced and upgraded in order to maximize the usage of the resources especially when the node is using the battery power sources. FIGURE 2.11 in Chapter 2 summarizes the flow of the willingness calculation, which includes the Core OLSR function and the power status plug-in on OLSR. In the basic OLSR core, only the default and manual willingness setting for the node itself is available. On the

other hand, the power status plug-in is where the willingness calculations are performed. The plug-in also has the function as the source of the power information that will be broadcasted into the network where the information will be picked by other node in order to know the power status of the other nodes.

The default value that is set in the core OLSR implementation is 3 because the information from the /proc/apm file is not available. The node may use the power source either using the AC or battery power. In this case, other nodes could not know the status and information about the node and for safety purpose; the willingness is set to the value of 3 because if the node uses the weak battery power, the network would automatically lose it without any warning invariably affecting the network to lose data. Alternatively, the setting can be based on user needs if they do not want to install power status plug-in with set the willingness to default or set the value (1-3 value) at the olsrd.conf file in order to allow the information for the node of the power status known to other devices. The willingness calculation on the other hand is located at the power status plug-in. The information flows to the power status plug-in function and then it will be calculated. The value of the willingness will be calculated when the usage of the AC power sources is 6. While the willingness value 1-3 indicating the percentage of balance of the battery power source. The values that have been calculated are not efficient because the resource is not consumed maximally. However, it has to maximize the usage of the power sources particularly in emergency situations such as natural disaster, fire, and earthquake and also for the military operations.

Therefore, it is proposed that the value is change from 6 (old willingness value) to 7 (new willingness value) when using AC power because the source can get the power consistently. When using the battery power source, the value change from 1 - 3 (old willingness value) to 0 – 7 (new willingness value), thus increasing the granularity of the possible values and maximizing the resources available to the nodes. The main reason for optimizing the power resource and willingness are to make sure all nodes have enough battery power resource in which the battery is limited and it have more chances to be selected as MPR. The details of the MPR are summarized in the FIGURE 3.1 below.

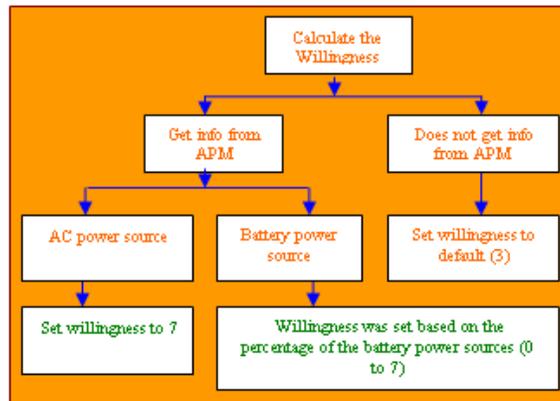


FIGURE 3.1: The new value for the willingness calculation flow diagram

3.3. OLSR INSTALLATION WITH POWER STATUS PLUG-IN SOFTWARE.

In default, the willingness value is automatically derived from the core OLSR software without the calculation of the willingness. However, it does not use the power status information from the /proc/apm file. It only uses either the default value that have been set or the value from the user setting. The willingness calculation function is located at the power status plug-in where the plug-in is not installed by default with the OLSR software during the installation. The power status plug-in should be installed manually when the user needs to optimize power utilization with OLSR. The OLSR software can work well if the power status plug-in is not installed and A/C power is used, but the results of such OLSR experiments are not highly reliable because it does not replicate the real situation. In real ad hoc network, sometimes there is no infrastructure and it needs the power status function to calculate the willingness based on the information from the /proc/apm file [8].

4.0 SETUP AND INSTALLATION

4.1. INTRODUCTION

The OLSRd software started out as part of the Master thesis project for Andreas Tonnesen at the University Graduate Centre. The OLSRd project is still on-going while the Master thesis has reached its completion. All nodes must be capable to detect and communicate with each other to establish a network where the infrastructure does not exist or where the services are not required. The IEEE 8102.11 Ad-hoc mode does not support multihop communications, whereas OLSR does. In the OLSR testbed, the Acer node could talk to Telco3 node through Telco2 node where the Acer node and Telco3 node were out of range (radio). The FIGURE 4.1 illustrates the testbed implemented in the experiment.

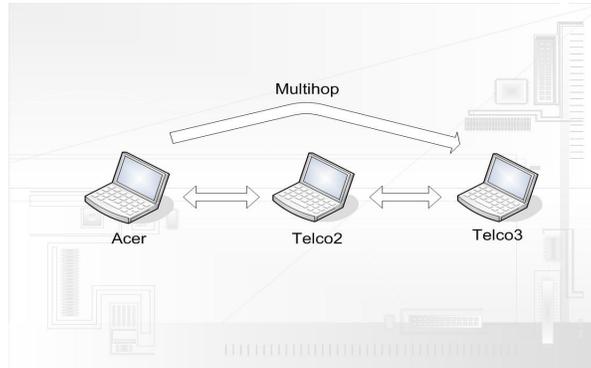


FIGURE 4.1: Multihop on OLSR

4.2. OLSR INSTALLATION

The nodes of the OLSR testbed were setup using Fedora Core 5 and Fedora Core 6 with the latest update kernel 2.6.19. It involved four notebooks that were installed with the OLSR software. The OLSR software can be downloaded through the internet at www.olsr.org web site. The source files were tarred using tar and bziped using bzip2. This experiment used the release source which was 0.4.10. Then, after extracting the file, it was compiled and installed after browsing and achieving it. After the compilation and installation was complete, the `olsrd.conf` file, which is located at `/etc/olsrd.conf` was configured. All nodes are configured with the static IP address (and they must be, since there is no DHCP server to provide a dynamic address).

4.3. NETWORK CONFIGURATION ON OLSR

OLSR nodes must get the latest update kernel of the Linux operating system (Fedora Core 5 and 6) from the ftp site and it also must be installed with gcc. The gcc in Linux is used to compile the source code after performing the changes of the OLSR source code. The wireless interface on each node should be configured with an IP address. The nodes should be configured with the same ESSID. The wireless card should be configured with "ad-hoc" mode. The other thing that is compulsory is to release the UDP/698 from blocking. When all the networks have been configured, the next step is to configure the `olsrd.conf` file. The main option that has to change is the debug level (0 – 9) which can be seen through the process where the daemon runs. Next is the IP version, and in this case it uses IPv4. Lastly, it will configure the wireless interface under the name "eth1". More details about the configuration can be found in the `olsrd.conf` document. The configuration can be enhanced depending on the needs of a user. The basic configuration needs on the `olsrd.conf` file is shown in Figure 4.2. The interfaces for the wireless are different for each other ensuring that the interface, which was already set at the `olsrd.conf` file, is correct. Then, the interface could be checked by using the command code at the terminal windows. Therefore, the tests that were planned performed well representing a small scale of the network (FIGURE 4.3).

```

Acer / Telco 1 / Telco2 / Telco3
/etc/sysconfig/network_scripts/ifcfg-eth1
...
MODE=Ad-Hoc
ESSID=olsr
CHANNEL=1
RATE=11M

/etc/olsrd.conf
DEBUG      3
IPVERSION  4
...
INTERFACE "eth1"
{...
Ip4Broadcast      255.255.255.255
...
}
    
```

FIGURE 4.2: The network configuration and olsrd.conf file for OLSR nodes.

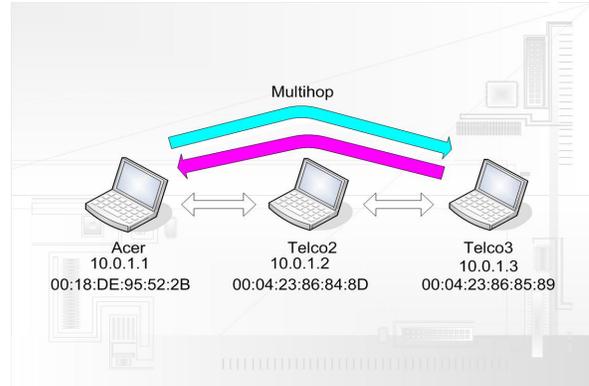


FIGURE 4.3: OLSR network communicate

On the other hand, the OLSR network can also be connected to the hard-wired (Ethernet) connection for a large network as well as network interface running on OLSR. It uses the Node and Network Association (HNA) message that contains sufficient information for the recipients to construct an appropriate routing table enabling the HNA (for IPv4 or IPv6) in the olsrd.conf file (FIGURE 4.5). FIGURE 4.4 illustrates the OLSR connection with the large fix connection.

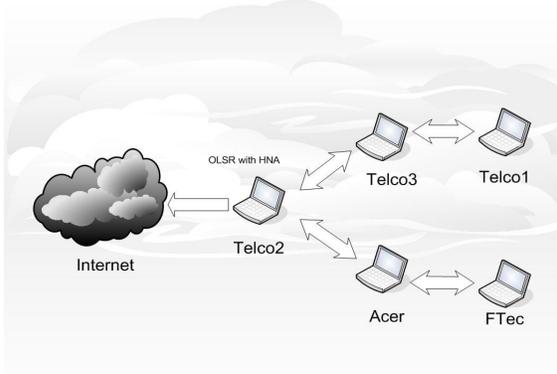


FIGURE 4.4: OLSR connection with large fix network.

```

Acer / Telco2 / Telco3
/etc/olsrd.conf
DEBUG      3
IPVERSION  4
...
Hna4
{
...
}
...
INTERFACE "eth1"
{
...
Ip4Broadcast      255.255.255.255
...
}
    
```

FIGURE 4.5: The HNA configuration on olsrd.conf file.

4.4. INSTALLATION THE PLUG-IN ON OLSR

The OLSR nodes were actually connected to each other without knowing the power status of the default node. It is actually optional for the OLSR to work without the power status function. In this experiment, the power status plug-in that has been provided with the OLSR source code file would be used during the downloading and installing process into the nodes. On the other hand, this function was not installed into the node when installing the OLSR automatically. Hence, it must be installed manually because it is optional for the OLSR as it is not a core function. The Power Status plug-in locates the *olsrd library* directory when the OLSR is installed into the computer. It uses the **Make** and **Make Install** commands to compile and install the power status function. When the compilation and the installation of the OLSR are complete, it must configure the olsrd.conf file as shown in Figure 4.6 below.

```

/etc/olsrd.conf
DEBUG      3
IPVERSION  4
...
LoadPlugin "/usr/lib/olsrd_power.so.0.3"
{
}
...
INTERFACE "eth1"
{...
Ip4Broadcast      255.255.255.255
...}

```

FIGURE 4.6: Adding the plug-in script olsrd.conf file

The “olsrd_power.so.0.3” would be located at the /usr/lib/ after the power status installation. It should be located at the correct path because it would show the error and the OLSR automatically terminates (stops) when it runs the OLSR. It must also have the open and close curly bracket after the plug-in function. Once completed, the OLSR network should be tested to determine whether an error has occurred or not.

5.0 NETWORK ANALISYS

When all the modifications were completely compiled, we next tested the resulting version on the network and collected some result from the OLSR deployment. The data collection only focuses on the availability of the nodes and also on the routing of the nodes on the ad-hoc network. This is because, the focus of the modification is to make sure the node on the OLSR is selected as MPR and to make sure the node is transferring the data willingly on the ad-hoc networks by using the limited battery power sources. Theoretically, the original OLSR source is calculating the percentage of the battery power balance and the willingness that can be acquired as summarized in FIGURE 5.1 and the modification of the OLSR source is summarized in FIGURE 5.2 as below.

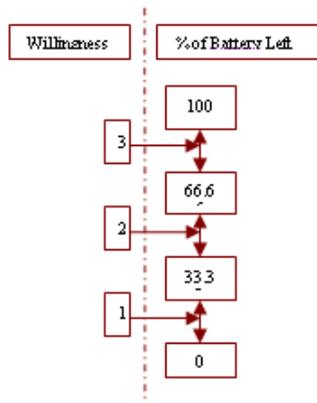


FIGURE 5.1: The percentage (%) of battery power balance with the willingness assign (original source)

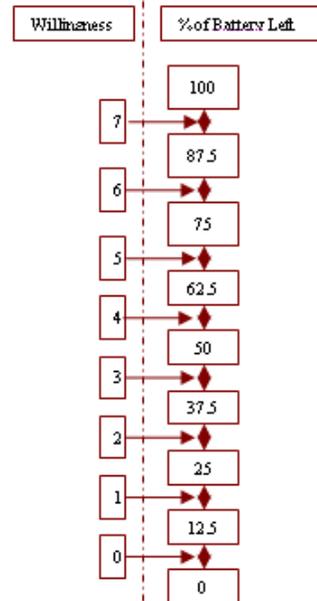


FIGURE 5.2 : The percentage (%) of battery power balance with the willingness assign (modification source)

Based on FIGURE 5.1 and FIGURE 5.2, comparison of the “Willingness” between the sources is illustrated in TABLE 5.1. The “Willingness” value based on the percentage (%) of battery power left is shown in Figure 5.7.

% of Battery Left	Willingness for Modify Source	Willingness for Original Source
100	7	3
90	7	3
80	6	3
70	5	3
60	4	2
50	4	2
40	3	2
30	2	1
20	1	1
10	0	1
0	0	0

TABLE 5.1: The willingness comparison table between original and modification sources

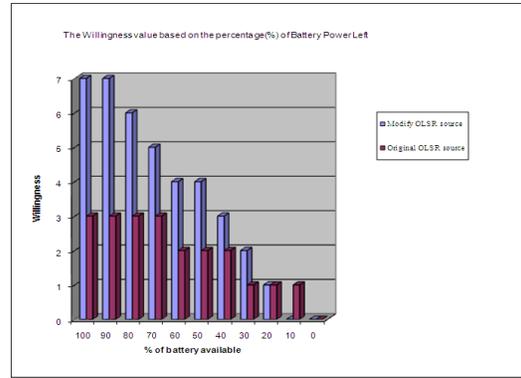


FIGURE 5.3: The comparison table on the willingness for original and modification sources based on the percentage of battery power left.

FIGURE 5.3 shows that the willingness for the modified source code is better than the original source code until the percentages of the battery power available reached 20% level. Then, the percentages of battery dropped from 12.5% to 0%, where the “Willingness” was set to 0, in order to give the node enough time to shutdown properly saving all the works being performed. TABLE 5.1 and FIGURE 5.3 show that the “Willingness” on the OLSR plays an important role in choosing a node as MPR.

OLSR availability

During the experiment on the OLSR ad-hoc network, the network itself must be available. It will affect the OLSR ad-hoc network if the nodes on that network were not available. It will also become a major factor in the measurement performance for the OLSR ad-hoc network. To test the availability of the nodes, “Ping” is used to check either the nodes are available or not on the network. In this experiment, four notebook computers were used for checking the OLSR availability. The details on the hardware and software specifications have been discussed in CHAPTER 3, in which the nodes setup is illustrated in FIGURE 5.4 overleaf.

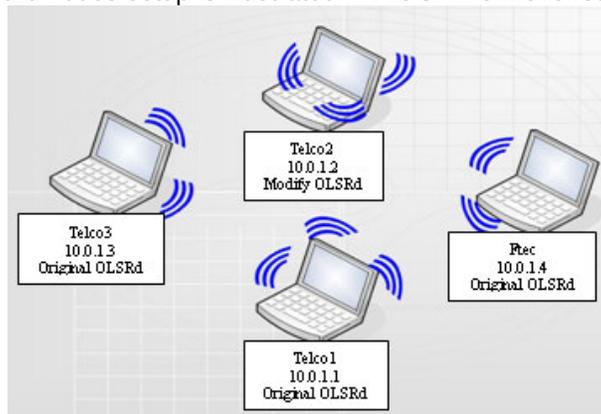


FIGURE 5.4: OLSR networking topology

5.1. USING AVAILABILITY STATISTIC DURING OLSR EXPERIMENT

When multiple ping packets are sent to a remote node, the ping program tracks on how many responses are received. The result will display as the percentage of the packets, which is not received. A network performance tools can use the “ping” statistic to obtain basic information regarding the status of the networks between the two end points. Once it is confirmed that there are lost packets in the “ping” sequence, it must determine the cause of the packet loss either due to the collisions on network segment or the packet being dropped by a network device. In the experiment, there was no packet loss on the OLSR networks.

5.2. ROUTING TABLE

When the network in all the OLSR nodes were available and all of them were communicating each other, the Telco3 node then moved far away from Ftec node until the node was out of range of the Ftec node. This is because the Telco3 node must use either the Telco1 or the Telco2 node in transferring the data to the Ftec node and vice versa. In the experiment, the setup of the nodes was performed as illustrated in FIGURE 5.16.

1. Ftec node (10.0.1.4) was located at the wireless room in the IT lab. Telco2 node (10.0.1.2) and Telco1 node (10.0.1.1) were located in the IT Lab near the door. All the three nodes were on the first floor situated on the left side of Wing C.
2. Telco3 node (10.0.1.3) was located on the second floor, in front of the lecture rooms, which was far away from Ftec node (10.0.1.4). Both nodes were out of line-of-sight range from one another. In this case, both nodes could only communicate via Telco1 or Telco2 node.

However, it needed to collect the data for analysis from the routing table for Ftec node and Telco3 node using a "traceroute" for Fedora Core 5 and "tracert" for Fedora Core 6. This experiment was performed based on time frame to get better result. The result shows that the Ftec node (source node) chose Telco2 node to transmit the data to the Telco3 node (destination) and the Telco3 node (source node) chose Telco2 node to transmit the data to Ftec node (destination node). Therefore, Telco2 node was selected as MPR because the value of the willingness was higher than telco1.

5.3. SUMMARY FROM THE EXPERIMENT

The result shows that the willingness plays an important role in choosing a node as MPR in OLSR Ad Hoc network. The higher value of the willingness of the node, the more possibility of that node to be chosen as MPR. The lower value of the willingness of the node, the lower the possibility of the node to be chosen as MPR. On the other hand, if all nodes have almost the same of the less value, then the MPR will be chosen according to the highest value among the nodes. However, the results will be different if the experiment involves more nodes and in different situations. There are also other factors that can generate different outcomes such as the distance between the nodes, the barriers in term of the radio frequency from the buildings, thick walls, mirrors, doors and others. In conclusion, the experimental results provide evidence that enhancing the facilities of OLSR by creating new scheme on calculating the willingness; it will create more paths in the network for the node to choose the shortest path to the destination eventhough the battery power resource is limited.

6.0 DISCUSSION AND CONCLUSIONS

6.1. OLSR Installation

Several important aspects need to be addressed when the OLSR network on the Fedora Core 5 and Fedora Core 6 as revealed in the experiment. The wireless adapter must be properly installed and fully functioning to ensure smooth data transmission. The latest kernel version for the system has to be put in place with the essential software to run the OLSR namely gcc for compiling and GTK2.0 development libraries to compile the GUI front-end. The wireless network configuration has to change from the "managed" mode change to "ad-hoc" mode and the setting of the ESSID, CHANNEL and RATE must be performed in accordance with the new setup. Testing for compatibility with Linux Fedora Core 5 and 6 and also other requirements need to be performed in the installation of OLSR using the third party sources. A complete documentation for the OLSR network development on different platform is essentially to help developers or researchers to setup their OLSR network easily and effectively. Currently, there is no OLSR for Windows that works on IPv6. The third party software for OLSR did not perform very well such as the PDA device like iPAQ, which is a run under the Windows Operating System like Windows Mobile 5.0.

6.2. LIMITATION

There are some limitations in the experiment conducted. The preparation of the OLSR platform needs each node to be wireless. The experiment was prepared for mobile ad-hoc network nodes requiring the deployment of at least four nodes, which were divided into 3 parts. First, it needs one source node, one destination node and at least two nodes in the middle network. The more

nodes used for the experiment, the better results obtained in making comparison with networks having less nodes will shows of that network in which it can find the disadvantages from that networks. It was quite difficult to get sufficient mobile nodes (notebooks) when performing the experiment due to the prohibitive cost. To fulfill the minimum requirements, available devices of the researcher and borrowed devices were used.

REFERENCES

1. C. Guillaume, & Eric F. (2002). *Ananas: A New Ad hoc Network Architectural Scheme*. INRIA Research Report – 4354.
2. Aleksandr H. (2004). *Comparing AODV and OLSR Routing Protocols*. Telecommunication Software and Multimedia Laboratory, Helsinki University of Technology. Sjukulla.
3. Laoutti, Muhlethaler P., Najid A., & Plakoo E. (2002). Simulation Results of the OLSR Routing Protocol for Wireless Network. *1st Mediterranean Ad-Hoc Networks workshop (Med-Hoc-Net)*. Sardegna, Italy.
4. Viennot L. (1998). *Complexity Results on Election of Multipoint Relays in Wireless Networks*. Technical Report, INRIA.
5. Jacquet P., Laouitti A., Minet P., & Viennot L. (2001). Performance Analysis of OLSR Multipoint Relay Flooding in Two Ad-Hoc Wireless Network Models. Research Report – 4260. *RSRCP Journal Special Issue on Mobility and Internet*. INRIA.
6. Jacquet P., Minet P., Muhlethaler P., Rivierre N. (1996). *Increasing Reliability in Cable Free Radio LANs: Low Level Forwarding in HIPERLAN*. Wireless Personal Communications.
7. Andreas T. (2004). *Implementing and Extending the Optimized Link State Routing Protocol*. Department of Informatics, University of Oslo. Oslo.
8. Clausen T., Jacquet P. (2003). Request for Comment: 3626. *Experimental Project Hipercom*.