Final Report of Project 4: Routing Protocols in mobile ad hoc networks

Shi Qiang, Wen Xiao, Guo Anjin, Yang Yingfeng

June 29th , 2010

1 Optimized Link State Routing

1.1 Introduction to OLSR

The Optimized Link State Routing (OLSR) is a table-driven, proactive routing protocol developed for MANETs. It is an optimization of pure link state protocols in that it reduces the size of control packet as well as the number of control packets transmission required.

OLSR reduces the control traffic overhead by using Multipoint Relays (MPR), which is the key idea behind OLSR. A MPR is a node's one-hop neighbor which has been chosen to forward packets. Instead of pure flooding of the network, packets are just forwarded by a node's MPRs. This delimits the network overhead, thus being more efficient than pure link state routing protocols. OLSR is well suited to large and dense mobile networks. Because of the use of MPRs, the larger and more dense a network, the more optimized link state routing is achieved.

MPRs helps providing the shortest path to a destination. The only requirement is that all MPRs declare the link information for their MPR selectors (i.e., the nodes who has chosen them as MPRs). The network topology information is maintained by periodically exchange link state information. If more reactivity to topological changes is required, the time interval for exchanging of link state information can be reduced.

1.2 Control messages

OLSR makes use of "Hello" messages to find its one hop neighbors and its two hop neighbors through their responses. The sender can then select its multipoint relays (MPR) based on the one hop node that offers the best routes to the two hop nodes. Each node has also an MPR selector set, which enumerates nodes that have selected it as an MPR node. OLSR uses Topology Control

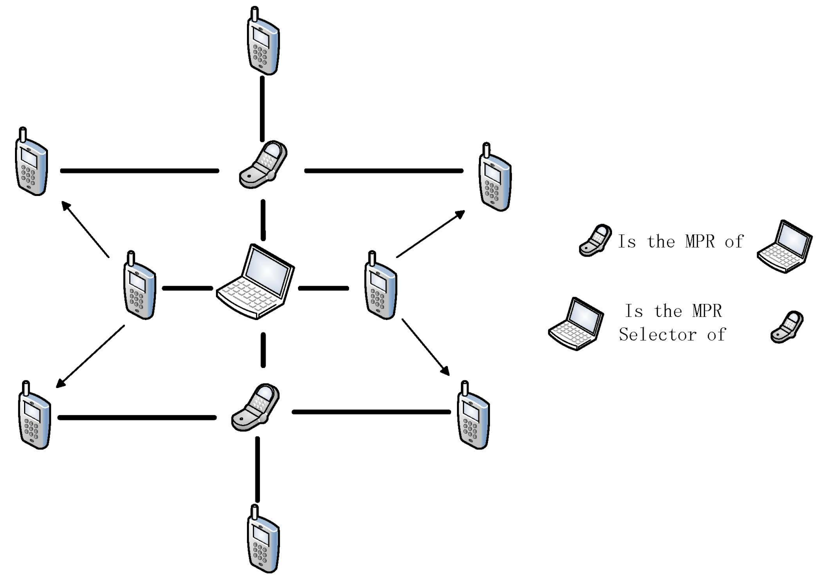


Figure 1: MPRs and MPR selector in OLSR network

(TC) messages along with MPR forwarding to disseminate neighbor information throughout the network. Host and Network Association (HNA) messages are used by OLSR to disseminate network route advertisements in the same way TC messages advertise host routes.

OLSR with a gateway (GW), that sends out HNA messages. All the other nodes may then be accessing the "Internet".

<table>
<tr><td colspan="10">0</td><td colspan="10">1</td><td colspan="10">2</td><td colspan="2">3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="16">Reserved</td><td colspan="8">Htime</td><td colspan="8">Willigness</td></tr>
<tr><td colspan="8">Link Code</td><td colspan="8">Reserved</td><td colspan="16">Link Message Size</td></tr>
<tr><td colspan="32">Neighbor Interface Address</td></tr>
<tr><td colspan="32">Neighbor Interface Address</td></tr>
<tr><td colspan="32">..</td></tr>
<tr><td colspan="8">Link Code</td><td colspan="8">Reserved</td><td colspan="16">Link Message Size</td></tr>
<tr><td colspan="32">Neighbor Interface Address</td></tr>
<tr><td colspan="32">Neighbor Interface Address</td></tr>
</table>

Figure 2: Hello message of OLSR

<table>
<tr><td colspan="10">0</td><td colspan="10">1</td><td colspan="10">2</td><td colspan="2">3</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>0</td><td>1</td></tr>
<tr><td colspan="16">ANSN</td><td colspan="16">Reserved</td></tr>
<tr><td colspan="32">Advertised Neighbor Main Address</td></tr>
<tr><td colspan="32">Advertised Neighbor Main Address</td></tr>
</table>

Figure 3: Topology Control message of OLSR

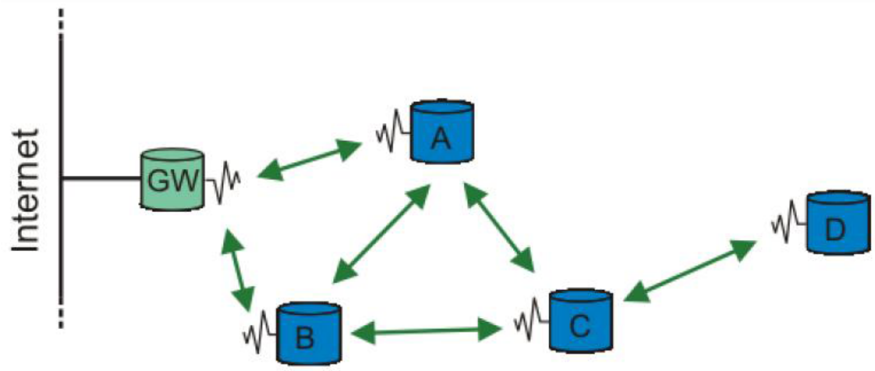


Figure 4: OLSR using GW

There's another message named Multiple Interface Declaration(MID), which is used for announcing that a node is running OLSR on more than one interface. The MID message is flooded throughout the network by the MPRs.

1.3 Neighbor discovery

As links in a ad hoc network can be either unidirectional or bidirectional, a protocol for determining the link status is needed. In OLSR, HELLO messages serve, among others, this purpose. HELLO messages are broadcasted periodically for neighbor sensing. When a node receives a HELLO message in which it's address is found, it registers the link to the source node as symmetric.

As an example of how this protocol works, consider two nodes A and B which not yet have established links with each other. First, A broadcasts an empty HELLO message. When B receives this message and does not find its own address in it, it registers in the routing table that the link to A is asymmetric. Then B broadcasts a HELLO message declaring A as an asymmetric neighbor. Upon receiving this message and finding its own address in it, A registers the link to B as symmetric. A then broadcasts a HELLO message declaring B as a symmetric neighbor, and B registers A as a symmetric neighbor upon reception of this message.

1.4 Topology information

Information about the network topology is extracted from topology control (TC) packets. These packets contain the MPRS set of a node, and are broadcasted by every node in the network, both periodically and when changes in the MPRS set is detected. The packets are flooded in the network using the multi-point relaying mechanism. Every node in the network receives such TC packets, from which they extract information to build a topology table.

1.5 Route calculation

When calculating a routing table for us, pure RFC-compliant OLSR simply minimizes the number of hops between ourselves and the other nodes in the MANET, even if this means that a route via a single very bad link will be preferred to a route via two excellent links, although the latter would probably have been the better choice.
To solve this problem, we have to teach olsrd how to tell good links from bad links. We have done so by measuring the packet loss for OLSR packets that we receive from our neighbors. As we periodically receive HELLO messages from our neighbors (by default every 2 seconds), we have enough packets to determine the packet loss for packets that each of our neighbors sends to us.

If, for example, 3 out of 10 packets are lost on their way from our neighbor to ourselves, we have a packet loss of $3/10 = 0.3 = 30\%$. At the same time 7 of the 10 packets that the neighbor sent went through. Hence, the probability for a successful packet transmission from this neighbor to ourselves is $7/10 =$

0.7 = 70%. This probability is what we call the Link Quality. So the Link Quality says how good a given link between a neighbor and ourselves is in the direction from the neighbor to ourselves. It does so by saying how likely it is that a packet that we send is successfully received by our neighbor.

However, it is also important to know the quality of the link in the opposite direction, i.e. how many of the packets that we send out are received by each of our neighbors. So, we are not only interested in the Link Quality of a given link, but also in the corresponding neighbor's idea of the Link Quality. That's what we call the Neighbor Link Quality. The Neighbor Link Quality says how good a given link between a neighbor and ourselves is in the direction from ourselves to the neighbor.

The Link Quality and the Neighbor Link Quality are values between 0 and 1 or, which is equivalent, between 0 and 100%. They represent the probability that a packet that our neighbor sends actually makes it to us (Link Quality) and that a packet that we send actually makes it to our neighbor (Neighbor Link Quality).

Let's now look at the probability for a successful packet round trip, i.e. the probability that we successfully send a packet to our neighbor and, on receiving it, our neighbor successfully replies with a response packet. For a successful round trip both packets must get through, the packet that we've sent and the response packet that our neighbor has sent. So, the success probability is $NLQ \times LQ$, where NLQ is the Neighbor Link Quality of the link and LQ is its link quality. For example, if we have a NLQ of 60% and a LQ of 70%, the probability of a successful round trip is $60\% \times 70\% = 0.6 \times 0.7 = 0.42 = 42\%$.

In wireless networks each recipient of a packet acknowledges packet reception by sending back an acknowledgment packet to the sender. So, when does a retransmission of a packet happen? It happens, if the sender does not receive an acknowledgment. And in which cases does the sender not receive an acknowledgment? If either the packet that it sent is lost or if the corresponding acknowledgment packet is lost. So, what is the probability for a retransmission to not take place? Well, as the sender's packet has to get through in one direction and the recipient's acknowledgment has to get through in the opposite direction, too, this is exactly the probability for a successful packet round trip, i.e. $NLQ \times LQ$.

We can now answer the question of how many transmission attempts it will typically take to get a packet from us to a neighbor or from the neighbor to us.It is $1/(NLQ \times LQ)$. So, in the above case of $NLQ \times LQ = 42\%$, we expect on average $1/0.42 = 2.38$ transmission attempts for a packet until it gets through.

Note that this number is valid for both directions of the link, as in both cases we have to look at the probability for a successful packet round trip. For packets that we send to our neighbor, the packet goes from us to the neighbor and the acknowledgment travels the other way around. For packets that our neighbor sends to us, the packet goes from the neighbor to us and the acknowledgment travels from us to the neighbor. In both cases a packet is sent in each direction and retransmission occurs if either packet is lost.

The value $1/(NLQ \times LQ)$ is called the Expected Transmission Count or

ETX. For those interested in a more in-depth discussion, there's a scientific paper by the people who invented all this, and for those who would like to know still more, there's Doug's PhD thesis.

Let's assume that we have a route from ourselves via two nodes A and B to a node C. What is the ETX for the total route, i.e. how often is our packet retransmitted on its way from us to C? Well, we know how many attempts we need on average to successfully transmit a packet from us to A. Let's call this value ETX1. So, we already have ETX1 attempts just to reach A. The packet would then take an additional number of attempts to hop from A to B. Let's call this second value ETX2. Finally, a further number of attempts is required to hop from B to C. Let's call this third value ETX3. Let's now have a look at the total number of transmissions that have happened to get our packet from us to C. This number is simply $ETX1 + ETX2 + ETX3$.

2 Comparison of AODV and OLSR

2.1 overview

Being a proactive protocol, OLSR imposes large control traffic overhead on the network. Maintaining an up-to-date routing table for the entire network calls for excessive communication between the nodes, as periodic and triggered updates are flooded throughout the network. The use of MPR's decrease this control traffic overhead, but for small networks the gain is minimal. The traffic overhead also consumes bandwidth.

The reactiveness of AODV is more sensitive to resource usage. As control traffic is almost only emitted during route discovery, most of the resource and bandwidth consumption is related to actual data traffic.

2.2 Resource usage

Because information about the entire network need to be maintained at all times, OLSR require relatively much storage complexity and usage. Hence, there is a greater demand for storage capacity of nodes in such networks. Also, the control overhead adds to the necessary processing in each node, hence increasing the battery depletion time. Another downside to OLSR is that it must maintain information about routes that may never be used, hence wasting possibly scarce resources.

AODV, on the other hand, only information about active routes are stored at a node, which greatly simplifies the storage complexity and reduces energy consumption. The processing overhead is also less than with OLSR, as little or no useless routing information is maintained.

2.3 Mobility

OLSR and AODV have different strengths and weaknesses when it comes to node mobility in MANETs. Unlike wired networks, the topology in wireless ad hoc networks may be highly dynamic, causing frequent path breaks to on-going sessions. When a path break occurs, new routes need to be found. As OLSR always have up-to-date topology information at hand, new routes can be calculated immediately when a path break is reported.

Because AODV is a reactive protocol, this immediate new route calculation is not possible, so a route discovery must be initiated.

In situations where the network traffic is sporadic, OLSR offers less routing overhead due to having found the routes proactively. AODV, on the other hand, will have to first discover a route before the actual information can be transmitted. This calls for extensive control overhead per packet. In cases where the network traffic is more or less static (i.e., the traffic has a long duration), however, AODV may perform better, as the amount of control overhead per packet decreases.

2.4 Route discovery delay

When a node in a network running the OLSR protocol wished to find the route to a host, all it has to do is do a routing table lookup, whereas in a AODV network, a route discovery process need to be initialized unless no valid route is cached. It goes without saying that a simple table-lookup takes less time than flooding the network, making the OLSR protocol performance best in delay-sensitive networks.

2.5 Link quality control and HNA extensions

For the OLSR link quality extensions, every link in the network is evaluated by LQ and route discovery is calculated by it. The route algorithm works out the best route concerning both hops and link quality factors using ETX, thus the route will be optimized. What's more, OLSR has a node acted as the gateway to the Internet and the usage of message HNA makes it possible to communication with other OLSR networks through Internet and wireless networks, that is, a OLSR cloud network comes true.

For the AODV protocol, route discovery is reactive and the link quality is labeled with reachable or unreachable. The link quality is assumed a binary thing, 0 or 1. So the route it discovered may not be optimized in terms of actual link quality factor. And the connectivity with other AODV networks via the Internet is yet to be implemented, which will be, however, not a simple work
.

2.6 Conclusion

AODV and OLSR both have distinctive characteristics which makes the one better suited than the other one, depending on the setting. As OLSR must maintain an up-to-date routing table at all times, a decrease in network performance is expected as more network overhead is needed. Most control overhead in AODV is related to route discovery, which is initiated when a path break occurs. In networks with low mobility, path breaks occurs less frequently, making AODV perform well.

OLSR will perform best when the traffic is sporadic, that is, when the traffic can benefit from having found a route proactively. This follows from that the single packet transmission delay is relatively small compared to running a route request protocol, as is done in AODV. For long duration traffic, however, AODV might perform better.

In networks with more or less static connectivity (i.e., little mobility), AODV performs best. The control overhead is kept at a minimum, so both bandwidth and energy consumption by control overhead is greatly reduced. These points make AODV more suited to resource and bandwidth critical situation.

3 Simulation of OLSR algorithm

3.1 Topology of a simple OLSR network

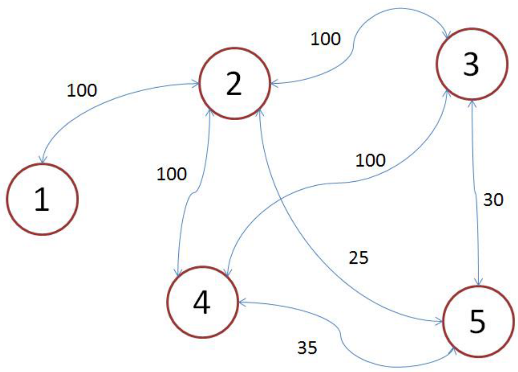


Figure 5: 5 nodes connected with the OLSR protocol

3.2 Message displayed in terminal 1

3.2.1 Links

```
--- 14:54:32.74 ---------------------------------------------- LINKS

IP address         hyst    LQ       lost      total   NLQ      ETX
10.0.0.2           0.000   1.250    0         10      0.243    3.29
```

The table contains the links to our neighbors. It contains the following columns:

- IP address - the IP address of the interface via which we have contact to the neighbor.
- hyst - the current hysteresis value for this link.
- LQ - the quality of the link determined at our end. This is what we have previously called the Link Quality.
- lost - the number of lost packets among the n packets most recently sent by our neighbor via this link. n is the link quality window size.
- total - the total number of packets received up to now. This value starts at 0 immediately after a link has come to life and then counts each packet. It is capped at the link quality window size.
- NLQ - this is our neighbor's view of the link quality. Previously we have called this the Neighbor Link Quality. This value is extracted from LQ HELLO messages received from our neighbors. NB: If a neighbor stops sending packets completely, we do not have any means of updating this value. However, in this case the LQ value will decrease and the link thus be detected as becoming worse.
- ETX - this is the ETX for this link, i.e. $1/(NLQ \times LQ)$.

3.2.2 Neighbors

```
--- 14:54:32.74 ---------------------------------------------- NEIGHBORS

IP address        LQ      NLQ     SYM    MPR    MPRS   will
10.0.0.2          1.250   0.243   YES    YES    NO     3

--- 14:54:32.02741652 ---------------------------- TWO-HOP NEIGHBORS

IP addr (2-hop)   IP addr (1-hop)   TLQ
10.0.0.3          10.0.0.2          0.018
10.0.0.4          10.0.0.2          0.018
10.0.0.5          10.0.0.2          0.199
```

The table contains a list of all our neibours. It is closely related to the link table in that we are connected to a neighbor via one or more links. The table has the following columns.

- IP address - the main IP address of the neighbor.
- LQ and NLQ - the LQ and NLQ values of the best link that we have with this neighbor. (In multi-interface configurations we can have more than one link with a neighbor.)
- SYM - this states whether the link to this neighbor is considered symmetric by olsrd's link detection mechanism.
- MPR (multi-point relay) - this indicates whether we have selected this neighbor to act as an MPR for us.

- MPRS (multi-point relay selector) - this indicates whether the neighbor node has selected us to act as an MPR for it.
- will - the neighbor's willingness.
- TLQ - the TLQ values of the best link that we have with this two-hop neighbor via the one-hop neighbor.

3.2.3 Topology

```
--- 14:54:32.74 ---------------------------------------- TOPOLOGY

Source IP addr   Dest IP addr     LQ     ILQ    ETX
10.0.0.2         10.0.0.3         0.239  0.243  17.19
10.0.0.2         10.0.0.1         0.239  0.243  17.19
10.0.0.2         10.0.0.4         0.239  0.243  17.19
10.0.0.2         10.0.0.5         0.875  0.875  1.31
10.0.0.3         10.0.0.2         0.239  0.243  17.19
10.0.0.3         10.0.0.4         0.239  0.243  17.19
10.0.0.3         10.0.0.5         0.114  0.749  11.74
10.0.0.4         10.0.0.2         0.239  0.243  17.19
10.0.0.4         10.0.0.3         0.239  0.243  17.19
10.0.0.4         10.0.0.5         0.114  0.749  11.74
10.0.0.5         10.0.0.2         0.624  1.000  1.60
10.0.0.5         10.0.0.3         1.000  1.000  1.00
10.0.0.5         10.0.0.4         1.000  1.000  1.00
```

This table displays the topology information that olsrd has gathered from LQ TC messages. It states which nodes in the network report links to which other nodes and which quality these links have. So, it's olsrd's view of the world beyond its immediate neighbor nodes, i.e. its view of the nodes that it cannot reach directly. This table has the following columns.

- Source IP addr - the node that reports a link.
- Dest IP addr - the node to which the source node reports the link.
- LQ (link quality) - the quality of the link as determined by the source node. For the source node this is the Link Quality. For the destination node this is the Neighbor Link Quality.
- ILQ (inverse link quality) - the quality of the link as determined by the destination node. For the source node this is the Neighbor Link Quality. For the destination node this is the Link Quality. We just did not want to name it "NLQ", as we use NLQ only for the link quality reported by our neighbors. But functionally this is equivalent to the NLQ we know from the link and neighbor tables.
- ETX - the ETX value for this link, calculated by $ETX = 1/(ILQ \times LQ)$.

3.2.4 Link control with olsr_switch(new control terminal)

The program, olsr_switch, is used to control the link state between two linked node within the OLSR networks.

```
root@wl-desktop:/home/wl/olsr# ./olsr_switch
olsrd host-switch daemon version 0.1 starting
Initiating socket TCP port 10150
OHS command interpreter reading from STDIN

 > link bi 10.0.0.1 * 0
Setting bidirectional link(s) 10.0.0.1 <=> 10.0.0.4
    quality 0
Setting bidirectional link(s) 10.0.0.1 <=> 10.0.0.3
    quality 0
Setting bidirectional link(s) 10.0.0.1 <=> 10.0.0.2
    quality 0

 > link bi 10.0.0.1 10.0.0.2 100
Removing bidirectional link(s) 10.0.0.1 <=> 10.0.0.2
    quality 100

 > list links
All configured links:
        10.0.0.4 => 10.0.0.1 Quality: 0
        10.0.0.3 => 10.0.0.1 Quality: 0
        10.0.0.1 => 10.0.0.3 Quality: 0
        10.0.0.1 => 10.0.0.4 Quality: 0

 > link bi 10.0.0.1 10.0.0.5 0
Setting bidirectional link(s) 10.0.0.1 <=> 10.0.0.5
    quality 0

 > list links
All configured links:
        10.0.0.5 => 10.0.0.1 Quality: 0
        10.0.0.4 => 10.0.0.1 Quality: 0
        10.0.0.3 => 10.0.0.1 Quality: 0
        10.0.0.1 => 10.0.0.5 Quality: 0
        10.0.0.1 => 10.0.0.3 Quality: 0
        10.0.0.1 => 10.0.0.4 Quality: 0

 > link bi 10.0.0.5 10.0.0.2 25
Setting bidirectional link(s) 10.0.0.5 <=> 10.0.0.2
    quality 25

 > link bi 10.0.0.5 10.0.0.3 30
Setting bidirectional link(s) 10.0.0.5 <=> 10.0.0.3
    quality 30

 > link bi 10.0.0.5 10.0.0.4 35
Setting bidirectional link(s) 10.0.0.5 <=> 10.0.0.4
    quality 35

 > list links
All configured links:
        10.0.0.5 => 10.0.0.4 Quality: 35
        10.0.0.5 => 10.0.0.3 Quality: 30
```

```
10.0.0.5 => 10.0.0.2 Quality: 25
10.0.0.5 => 10.0.0.1 Quality: 0
10.0.0.4 => 10.0.0.5 Quality: 35
10.0.0.4 => 10.0.0.1 Quality: 0
10.0.0.3 => 10.0.0.5 Quality: 30
10.0.0.3 => 10.0.0.1 Quality: 0
10.0.0.2 => 10.0.0.5 Quality: 25
10.0.0.1 => 10.0.0.5 Quality: 0
10.0.0.1 => 10.0.0.3 Quality: 0
10.0.0.1 => 10.0.0.4 Quality: 0
```

3.2.5 End a terminal

When a node is shutting down from the OLSR network. Message are displayed as following.

```
Received signal 2 - shutting down
Deleting all routes...
 HNA list:
Route list:
Dest: 10.0.0.5
Dest: 10.0.0.4
Dest: 10.0.0.3
Dest: 10.0.0.2
Closing sockets...
Closing plugins...
Restoring network state
```

Some of them are interpreted as:

Received signal 2 - shutting down when a node is leaving the OLSR network.

HNA represents for "Host and Network Association", which node works as a interface for other nodes in the wireless OLSR networks to link with another OLSR networks through Internet, thus forming a OLSR cloud networks.

Socket is a mechanism used to communicate within computer networks.

Plugin is the feature ,httpinfo for example, implemented in the olsrd program.

3.3 Analysis of one terminal

The text displayed in terminal 3 is as following

```
*** olsr.org - 0.5.3 (Jun  4 2010) ***

--- 14:54:54.47 ------------------------------------------ LINKS

IP address        hyst   LQ     lost   total  NLQ    ETX
10.0.0.5          0.000  0.750  4      10     0.875  1.52
10.0.0.4          0.000  1.250  0      10     0.243  3.29
```

```
10.0.0.2          0.000  1.250  0      10     0.243  3.29

--- 14:54:54.47 ---------------------------- NEIGHBORS

IP address        LQ     NLQ    SYM   MPR   MPRS  will
10.0.0.2          1.250  0.243  YES   YES   NO    3
10.0.0.4          1.250  0.243  YES   NO    NO    3
10.0.0.5          0.750  0.875  YES   YES   NO    3

--- 14:54:54.02471059 --------------- TWO-HOP NEIGHBORS

IP addr (2-hop)   IP addr (1-hop)  TLQ
10.0.0.2          10.0.0.5         0.244
                  10.0.0.4         0.018
10.0.0.1          10.0.0.2         0.073
10.0.0.4          10.0.0.5         0.409
                  10.0.0.2         0.018
10.0.0.5          10.0.0.4         0.171
                  10.0.0.2         0.199

--- 14:54:54.47 ---------------------------- TOPOLOGY

Source IP addr    Dest IP addr     LQ     ILQ    ETX
10.0.0.2          10.0.0.3         0.239  0.243  17.19
10.0.0.2          10.0.0.1         0.239  1.000  4.18
10.0.0.2          10.0.0.4         0.239  0.243  17.19
10.0.0.2          10.0.0.5         0.749  0.875  1.53
10.0.0.1          10.0.0.2         0.239  0.243  17.19
10.0.0.4          10.0.0.2         0.239  0.243  17.19
10.0.0.4          10.0.0.3         0.239  0.243  17.19
10.0.0.4          10.0.0.5         1.000  0.624  1.60
10.0.0.5          10.0.0.2         1.000  0.373  2.68
10.0.0.5          10.0.0.3         0.749  0.875  1.53
10.0.0.5          10.0.0.4         0.624  1.000  1.60
```

It's clearly acknowledged that the link quality of the LINKS and NEIBORS are measured up to 1.25 as 100%.The topology is the same for all the terminals in the OLSR networks, which concludes all the possible links in the network. As we can see, the link may be asymmetric between two nodes, thus making the LQ factor important when determining the optimized link between two nodes in the OLSR network. For node 3, node 2 and 5 act as its MPR and node 3 itself is not a suitable MPR for any node in the network according to ETX, that is, node 3 is not a MPRS for any node in the OLSR network.

4 Implementation of OLSR

4.1 Using IPV6

We have established a simple network with two computers using IPV6 wireless network in OLSR protocol. A test with more nodes and HNA feature is to be accomplished once the computers of the lab are available. Moreover, some interesting tests such as video transmission or data transmission and so on are also on the schedule. Here is what we have done:

4.1.1 Set up the IPV6 wireless network

On computer 1:

```
sudo su
insmod /lib/modules/2.6.16-001/kernel/drivers/net/
    wireless/rt61.ko
ifconfig ra0 up
iwconfig ra0 mode ad-hoc essid olsr enc off
ifconfig ra0 inet6 add 2002:5c0:8d03:1::1/64
iwconfig ra0 channel 1
echo "1" >/proc/sys/net/ipv6/conf/ra0/forwarding
echo "1" >/proc/sys/net/ipv6/conf/ra0/autoconf
echo "1" >/proc/sys/net/ipv6/conf/ra0/accept_ra
echo "1" >/proc/sys/net/ipv6/conf/ra0/accept_redirects
```

On computer 2, the work is similar except for changing the inet6 IP to $2002 : 5c0 : 8d03 : 1 :: 2/64$.

4.1.2 Set up OLSR work

On computer 1, after the OlSRD-0.5.3 is installed:

```
./olsrd -i ra0 -d 1 -ipv6
```

The config file is the default */etc/olsd.conf*. On computer 2, type the same command.

4.1.3 Partial results on computer 1 and 2

On computer 1:

```
      *** olsr.org - 0.5.3 (Jun  4 2010) ***

--- 17:23:42.56 ---------------------------------------- LINKS

IP address            hyst   LQ     lost    total  NLQ    ETX
2002:5c0:8d03:1::2    0.000  1.250  0       10     0.243  3.29
--- 17:23:42.56 -------------------------------------------- NEIGHBORS

IP address            LQ     NLQ    SYM   MPR   MPRS  will
2002:5c0:8d03:1::2    1.250  0.243  YES   NO    NO    3

--- 17:23:42.02567435 ----------------------- TWO-HOP NEIGHBORS

IP addr (2-hop)  IP addr (1-hop)  TLQ

--- 17:23:42.56 ---------------------------------------- TOPOLOGY

Source IP addr       Dest IP addr         LQ     ILQ    ETX
2002:5c0:8d03:1::2   2002:5c0:8d03:1::1   0.239  0.243  17.19
```

On computer 2:

```
--- 17:23:42.70 ---------------------------------------- LINKS

IP address            hyst    LQ      lost    total  NLQ     ETX
2002:5c0:8d03:1::1    0.000   1.250   0       10     0.243   3.29
--- 17:23:42.70 --------------------------------- NEIGHBORS

IP address            LQ      NLQ     SYM    MPR    MPRS   will
2002:5c0:8d03:1::1    1.250   0.243   YES    NO     NO     3

--- 17:23:42.02701647 ---------- TWO-HOP NEIGHBORS

IP addr (2-hop)  IP addr (1-hop)  TLQ

--- 17:23:42.70 ------------------------------------ TOPOLOGY

Source IP addr       Dest IP addr          LQ      ILQ     ETX
2002:5c0:8d03:1::1   2002:5c0:8d03:1::2    0.239   0.243   17.19
-
--- 17:23:56.48 ---------------------------------------------- LINKS
...
...
--- 17:24:06.62 -------------------------------------------- LINKS

IP address          hyst    LQ      lost    total  NLQ     ETX
2002:5c0:8d03:1::1  0.000   0.500   6       10     0.243   8.23
--- 17:24:06.62 -------------------------------------- TOPOLOGY

Source IP addr      Dest IP addr          LQ      ILQ     ETX
2002:5c0:8d03:1::1  2002:5c0:8d03:1::2    0.239   0.243   17.19

TC-SET: Deleting empty entry 2002:5c0:8d03:1::1 ->
...
...
--- 17:24:12.70 ---------------------------------------------- LINKS

IP address            hyst    LQ      lost    total  NLQ     ETX
2002:5c0:8d03:1::1    0.000   0.125   9       10     0.243   32.90


--- 17:24:12.70 -------------------------------------- TOPOLOGY

Source IP addr  Dest IP addr          LQ      ILQ     ETX
Deleting route: 2002:5c0:8d03:1::1
...
...
--- 17:24:19.51 ---------------------------------------- LINKS

IP address      hyst    LQ      lost    total  NLQ     ETX
```

Computer 1 is shut down shortly after 17:23:42.56, then we can see from results of computer 2 that nearly 14 seconds later it sets up link change(normally 2 seconds). The deleting process is as following: first begin deleting node 1 in TC message when LQ reduces to 0.500, then begin deleting it from the route when LQ reduces to 0.125 and finally delete it when LQ reduces to 0.000. The total deleting process lasts about 37 seconds. Due to that the block down of the link between two computer is unavailable using the iptable6 command, thus making every node of the IPV6 network is bi-connected to all other nodes. So the rest work of the topology is yet to be done in the future.

4.2 Using IPV4

We have successfully established an OLSR network with 7 nodes using IPV4. A small test of video transmission is done using the program VLC.

4.2.1 Set up the IPV4 wireless network

we establish the network using the method of lab3 in the wireless class.

```
ifconfig ra0 up
iwconfig ra0 essid test_group1 mode ad-hoc
ifconfig ra0 10.1.1.x netmask 255.255.255.0 up
```

4.3 Set up OLSR work

On every computer, after the OlSRD-0.5.3 is installed:

```
./olsrd -f /etc/olsrd.conf -i ra0 -d 1
```

4.3.1 Topology of the network

The network is made up of two OLSR subnetwork and the two is connected through the wireline work, thus making a cloud OLSR network. The unidirectional link between node 1 and node 4 means that node 1 can receive packets sent from node 4 and the reverse is not established.

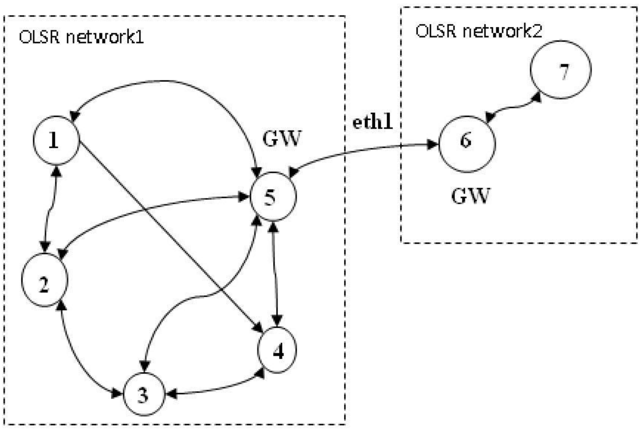


Figure 6: 7 nodes connected with the OLSR protocol

4.3.2 Information of the network

In the table below, we show the OLSR network of the node 1.

```
Kernel IP routing table
Destination     Gateway    Genmask          Flags Metric Ref Use Iface
10.1.2.2        10.1.1.5   255.255.255.255 UGH   2      0    0  ra0
10.1.2.1        10.1.1.5   255.255.255.255 UGH   3      0    0  ra0
192.168.1.100   10.1.1.5   255.255.255.255 UGH   2      0    0  ra0
192.168.1.117   10.1.1.5   255.255.255.255 UGH   1      0    0  ra0
10.1.1.5        0.0.0.0    255.255.255.255 UH    1      0    0  ra0
10.1.1.4        10.1.1.5   255.255.255.255 UGH   2      0    0  ra0
10.1.1.3        10.1.1.5   255.255.255.255 UGH   2      0    0  ra0
10.1.1.2        0.0.0.0    255.255.255.255 UH    1      0    0  ra0
```

The node 5 acts as the gateway of OLSR network1 and every node can be connected to OLSR network2 through it. And the gateway of node1 to node 3 is changing, that's, sometimes it's node 5 and sometimes node 3. It's a proof that the LQ parameter is working during the choice of a good route.
Then the link and topology information table of node 1 is shown as following:

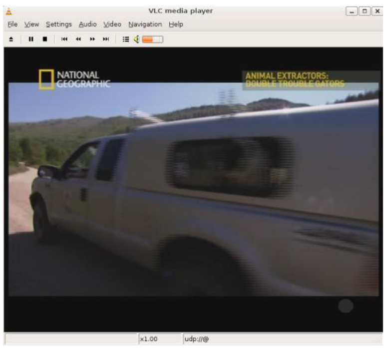
```
*** olsr.org - 0.5.3 (Jun 19 2010) ***

--- 03:47:09.59 ---------------------------------------- LINKS

IP address         hyst    LQ      lost     total  NLQ     ETX
10.1.1.2           0.000   1.000   2        10      0.243   4.11
10.1.1.5           0.000   1.250   0        10      0.243   3.29
10.1.1.4           0.000   1.250   0        10      0.000   0.00

--- 03:47:09.59 ------------------------------- NEIGHBORS

IP address         LQ      NLQ     SYM    MPR    MPRS   will
10.1.1.2           1.000   0.243   YES    YES    NO     3
10.1.1.4           1.250   0.000   NO     NO     NO     3
10.1.1.5           1.250   0.243   YES    YES    NO     3

--- 03:47:09.02598896 -------------- TWO-HOP NEIGHBORS

IP addr (2-hop)    IP addr (1-hop)   TLQ
10.1.1.2           10.1.1.5          0.018
10.1.1.3           10.1.1.2          0.014
                   10.1.1.5          0.009
                   10.1.1.4          0.000
10.1.2.2           10.1.1.5          0.018
10.1.1.4           10.1.1.5          0.018
10.1.1.5           10.1.1.2          0.014
                   10.1.1.4          0.000

--- 03:47:09.59 -------------------------------- TOPOLOGY

Source IP addr     Dest IP addr      LQ      ILQ     ETX
10.1.1.2           10.1.1.3          0.239   0.243   17.19
10.1.1.2           10.1.1.1          0.875   0.243   4.70
10.1.1.2           10.1.1.5          0.239   0.243   17.19
10.1.2.1           10.1.2.2          0.239   0.243   17.19
```

10.1.1.3	10.1.1.2	0.239	0.243	17.19
10.1.1.3	10.1.1.4	0.114	0.243	36.17
10.1.1.3	10.1.1.5	0.114	0.243	36.17
10.1.2.2	10.1.2.1	0.239	0.243	17.19
10.1.2.2	10.1.1.5	0.239	0.243	17.19
10.1.1.4	10.1.1.3	0.239	0.118	35.53
10.1.1.4	10.1.1.5	0.239	0.243	17.19
10.1.1.5	10.1.1.2	0.239	0.243	17.19
10.1.1.5	10.1.1.3	0.239	0.118	35.53
10.1.1.5	10.1.2.2	0.239	0.243	17.19
10.1.1.5	10.1.1.1	0.239	0.243	17.19
10.1.1.5	10.1.1.4	0.239	0.243	17.19

In the NEIGHBOR table, the link of node 1 to node4 has a good quality of LQ as 1.25 and a worst quality of NLQ as 0.00, which means the link is unidirectional as designed.

4.3.3 Video transmission test

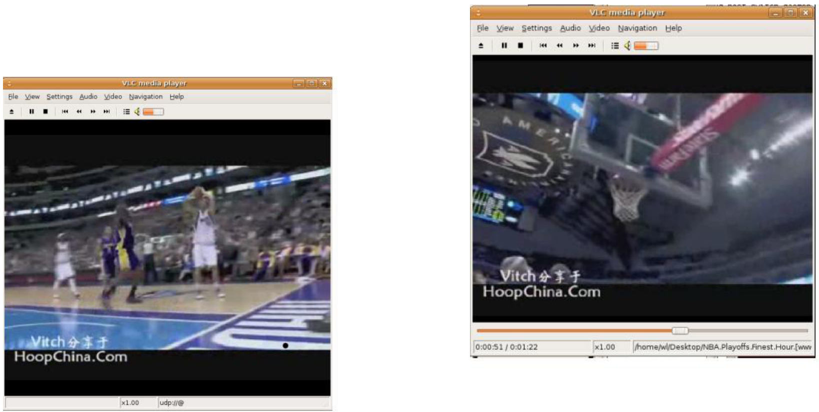
We have done a test of low definition and high definition video transmission between node 2 and node 7 shown in Figure 5, node 2 as sender and node 7 as receiver. The screen is obtained simultaneously.



(a) Vedio on 2

(b) Video on 7

Figure 7: High definition video transmission



(a) Vedio on 2

(b) Video on 7

Figure 8: Low definition video transmission

From the ping command on node 7 to node 2, it's known that the higher definition the video transmission is, the greater delay is, which is shown as following:

```
64 bytes from 10.1.1.2: icmp_seq=49 ttl=61 time=14905 ms
64 bytes from 10.1.1.2: icmp_seq=49 ttl=61 time=14905 ms (DUP!)
64 bytes from 10.1.1.2: icmp_seq=61 ttl=61 time=3233 ms
64 bytes from 10.1.1.2: icmp_seq=62 ttl=61 time=2233 ms
64 bytes from 10.1.1.2: icmp_seq=63 ttl=61 time=1230 ms
64 bytes from 10.1.1.2: icmp_seq=64 ttl=61 time=228 ms
64 bytes from 10.1.1.2: icmp_seq=65 ttl=61 time=2.94 ms
64 bytes from 10.1.1.2: icmp_seq=66 ttl=61 time=3.65 ms
64 bytes from 10.1.1.2: icmp_seq=67 ttl=61 time=2.63 ms
64 bytes from 10.1.1.2: icmp_seq=68 ttl=61 time=2.94 ms
64 bytes from 10.1.1.2: icmp_seq=69 ttl=62 time=4.06 ms
64 bytes from 10.1.1.2: icmp_seq=70 ttl=62 time=2.10 ms
64 bytes from 10.1.1.2: icmp_seq=71 ttl=62 time=1.92 ms
```

5 Conclusion

In this paper, we have discussed the OLSR protocol and compared it with the AODV protocol. A simulation within one computer of 5 nodes is done to test the OLSR algorithm and a simple real network within two computer using ipv6 is also successfully implemented, of which the results analysis available. And a more complex network of 7 nodes using IPV4, including the combination of two OLSR network using GW of OLSR, is also successfully established and an interesting test of video transmission has also been included.

Originally, we have set up two directions to simulate or establish a network using OLSR protocol: one is NS2 simulation and the other is the real LINUX implementation. Though the first direction meets a lot of obstacles and doesn't not meet our expectations, a lot of meaningful and useful work has been done and we have devoted ourself to it until the last time. The second direction is successfully established on the computers in the lab. The basic method and experiment have been carefully designed and several interesting tests of OLSR characters have been novelly done. More work on OLSR is yet to be done, like the optimization of the algorithm, in the future.

6 Appendix

6.1 Members of our group

Figure 9: Members of Group 14(From left to right): Yang Yingfeng, Wen Xiao, Guo Anjin, Shi Qiang

6.2 Contribution description

Guo Anjin: Oral presentation, paper preparation, lab assistant,group leader;
Yang Yingfeng:Oral presentation, paper preparation, lab assistant;
Wen Xiao: Oral presentation, paper preparation, lab assistant, NS2 simulation research;
Shi Qiang: Oral preparation, paper writing, OLSR lab design.

References

[1] Kenneth Holterna. Comparing aodv and olsr. 2005.

[2] olsr.org. Olsr_switch network simulation. 2005.

[3] olsr.org. Olsr link quality extensions. 2006.

[4] tldp.org. Linux optimized link state routing protocol (olsr) ipv6 howto. 2004.

[5] Xinbing Wang. Mobile ip. 2010.