A research on the IFQ of MPOLSR in NS2

Jiazi Yi

August 26, 2008

Abstract

This research is focused on the length of the interface queue (IFQ) of the MPOLSR protocol. The length of the IFQ in each node of the ad hoc network have big influence in the performance of the networks, both the data delivery ratio and the end-to-end delay. The full of the queue will result in packets loss and the congestion of the queue will cause longer delay. This work examined the behavior of the IFQ in the nodes and compared the performance in different protocols and configurations to give some clues about the IFQ in the future research.

1 Implementation of IEEE MAC 802.11 and the cross-layer information in NS2

For a single node, the simulation of the ad hoc networks based on IEEE 802.11 with NS2 has several components below:

- Channel: The WirelessChannel simulates the physical media (air) for wire less communication. It keeps a list of all nodes on this channel (mostly, all the nodes participate the simulation). The list is sorted based on the X-dimension values of nodes before it can be used.
- Network Interface: In the class WirelessPhy, NS2 simulates the wireless physical layer. Besides working for receiving and sending packets as any other physical layer, it deals with the propagation model (*free space*, *two ray ground* or *shadowing*), node sleeping (duty cycle management), energy model management and maybe different antenna models and modulations. It also has the class **MobileNode** to trace the mobility of nodes.
- Media Access Control (MAC):
- Outgoing Queue: The only target of the outgoing queue is MAC. It might has many variations such as droptail, priqueue, etc.
- Link Layer: It has a composed component ARP which works as ARP procedure, mapping the protocol address (such as IP address) to hardware address (such as MAC) address.

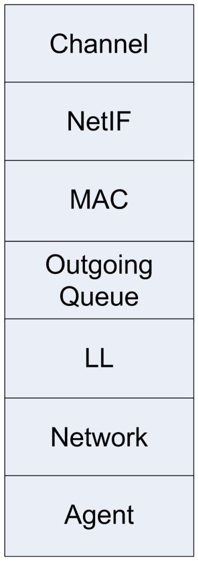

Figure 1: The architecture of the mobile node

- Network Layer: Routing agents with many variations such as DSDV, DSR, AODV, etc. The MPOLSR is also located in this layer.
- Agent Layer: Generates the data traffic in different manners, such as CBR or FTP.

The architecture is presented in Figure 1

Through the trace file of the NS2, we can monitor the behavior of the agent layer, network layer and the mac layer, such as below

To get the lower lever information for the network layer, we need to modify the code of the protocol in NS2 following the steps below:

- In MPOLSR.h, declare the Queue and Link Layer object

```
NsObject *ll; //link layer
CMUPriQueue * ifq; //outgoing queue
```

- In the command() of MPOLSR.cc, initiate the related object

```
...
TclObject *obj;
obj = TclObject::lookup(argv[2]==0);
ll = (NsObject*)obj;
obj = TclObject::lookup(argv[3])==0);
ifq = (CMUPriQueue*)obj;
```

- In the simulation script, add the following code:

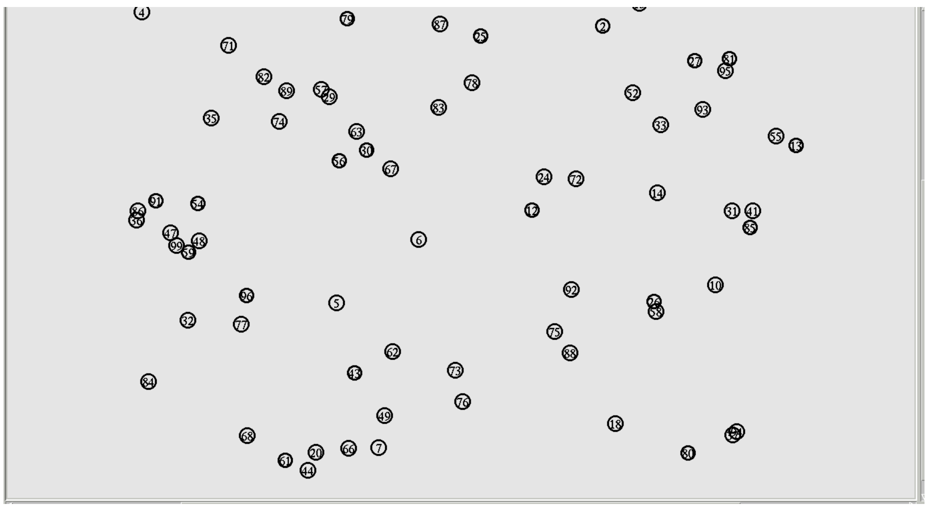

Figure 2: Network Topoloy

...

set rt($i) [$node_($i) agent 255];
$rt add_ll [$node_ ($i) set ll_(0)] [$node_($i) set ifq_(0)];

- And then we can get the lower layer method by calling the method of the lower layer objects. For example, to get the length of the queue, just use ifq->prq.length().

2 Simulation Environment

The simulation is taken in NS2. The number of the node is 50 in a 1000*1000m area, with no mobility. A pair of the source and destination are chosen for the data transmission, which is about 5 hops away.

The simulation lasts for 200 seconds, and source sends data from 30s to 170s. We chose the static scenario and single data source to make the problem easy to trace. The length of the queue at each node is recorded when it is trying to forward a data packet. In this way, the behavior of the length of the queue can be monitored. In this simulation, the max queue length is set to 50, which means that when the new arrival packets will dropped if the queue is full.

The source node is 47 and the destination is 37, as shown in Figure 2, which presents the part of the network topology related.

3 OLSR and the original MPOLSR

The UM-OLSR is used as the implementation of OLSR. The original MPOLSR uses the Multipath Dijkstra Algorithm to compute the multipath, and made

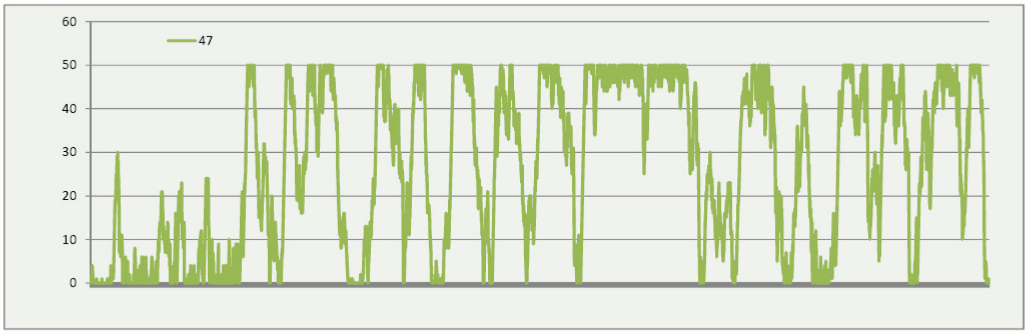

Figure 3: Queue length in node 47 at 100pkts/s, OLSR

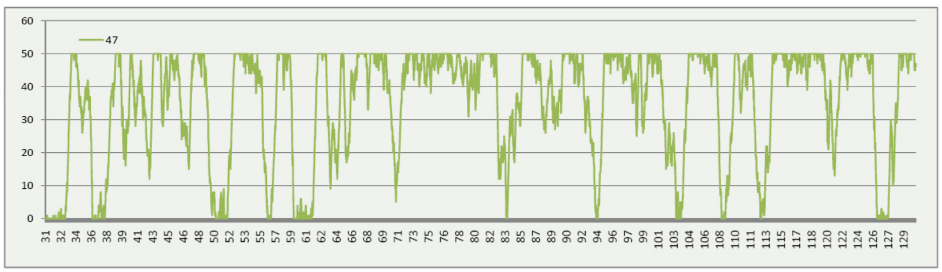

Figure 4: Queue length in node 47 at 100pkts/s, Ori-MPOLSR

two more modifications on the proactive nature of OLSR:

- Modification of TC message: in OLSR, the TC message just includes the link between local node and MPR nodes. In the MPOLSR we proposed before, the TC message includes the link between local node and all the neighbors to discover more links to construct the multipath.
- The neighborhood holding time in each node: in OLSR, the neighborhood holding time is 6 seconds in each node. And in the MPOLSR, we changed it to 4 seconds so that we can discover the link loss more quickly.

In fact, in the simulations we did before, those two modifications didn't make significant improvement for the performance. And in this study, when the data rate is very high, it will influence the delay of the networks, which will be discussed in detail later.

Figure 3 and Figure 4 show the length of the queue at the source node 47 in Ori-MPOLSR and OLSR. The source node data rate is set to 100pkts/s, and each packet is 512 bytes. As we can see from the figures, given the data rate as high as 100pkts/s, the queue will be saturated and result in packet loss and long delay, both for OLSR and Ori-MPOLSR.

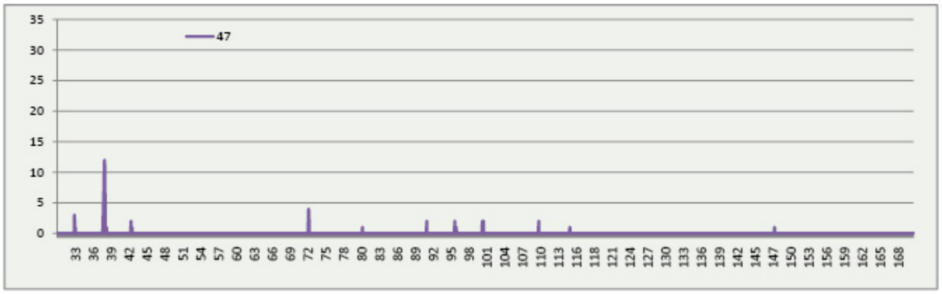

Figure 5: Queue length in node 47 at 40 pkts/s, OLSR

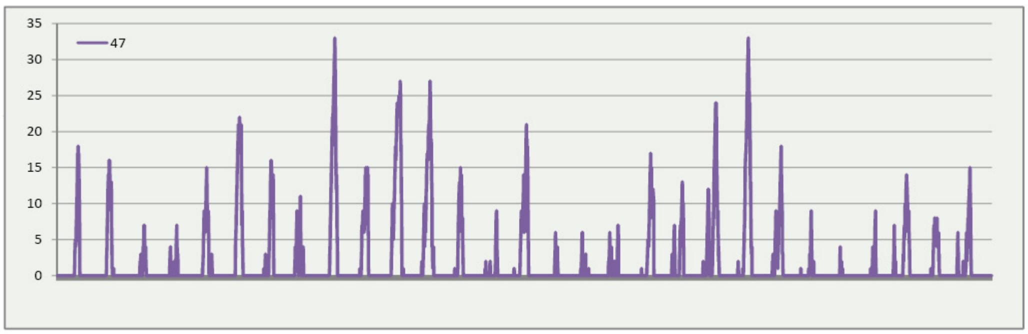

Figure 6: Queue length in node 47 at 40 pkts/s, Ori-MPOLSR

Figure 5and figure 6 show the length at the data rate of 40 pkts/s. At the source node, Ori-MPOLSR has much more congestion than OLSR at the source node, which also cause much longer delay (0.11275s for Ori-MPOLSR and 0.01829s for OLSR).

And if we go further and check the figures and trace file in detail, as shown in figure 7, which presents a case of congestion in the source node 47. We will find out that the congestion is mostly located in the first or the second hop and just last for one or two seconds. In the protocol, the HELLO message interval is 2 seconds and the TC message interval is 5 seconds. It means that it might be impossible for us to use the HELLO message and TC message to carry the information of the queue length to locate the redundancy of the MDC in this situation. Because when the information arrives, the real situation of the queue length has been changed.

To find out the reason of the longer delay of original MPOLSR, we did the simulation in different configurations.

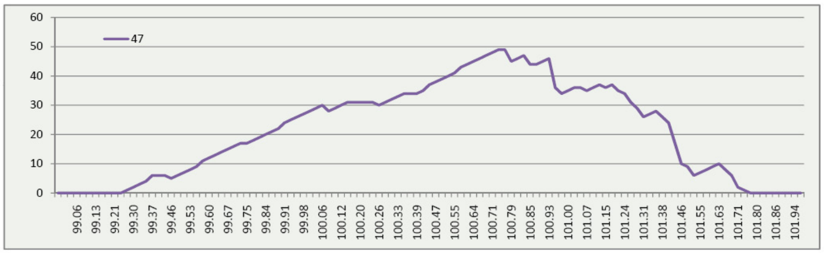

Figure 7: A case of congestion

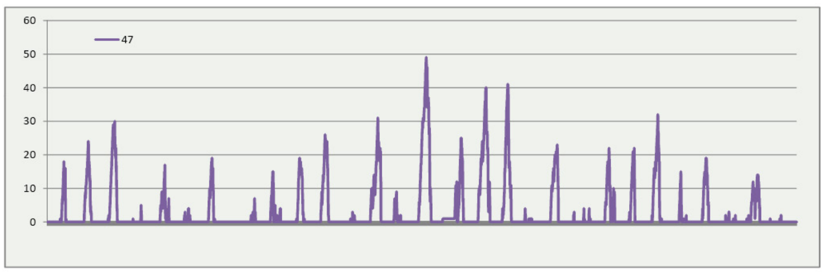

Figure 8: Queue length in node 47 at 40 pkts/s, single path for Ori-MPOLSR

4 Single path and multipath

Figure 8 presents the situation of the queue length at the source node using single path Ori-MPOLSR. Compared with figure 6, which uses the multipath, the single path protocol tends to have more congestion. And if we examine the queue length at the second hop (not presented here because of the space), we will find out that the single path protocol also has longer queue waiting than the multipath protocol. It also means that the multipath one can balance the load in this point of view. The impacts of the delay will be shown in the later discussion.

5 The impacts of neighborhood hold time

If we check the number of the sent routing packets, we'll notice that compared with OLSR, the Ori-MPOLSR is about 15% more. This is mainly because in the Ori-MPOLSR, the neighborhood holding time is changed from 6 seconds to 4 seconds.

This might change the sending of the TC messages. The interval between the transmission of two TC message depends upon whether the MPR Selector set is changed or not, since the last TC message transmitted. When a change occurs in the MPR Selector set, the next TC message may be sent earlier then the scheduled time. When the neighborhood hold time is shortened, it will

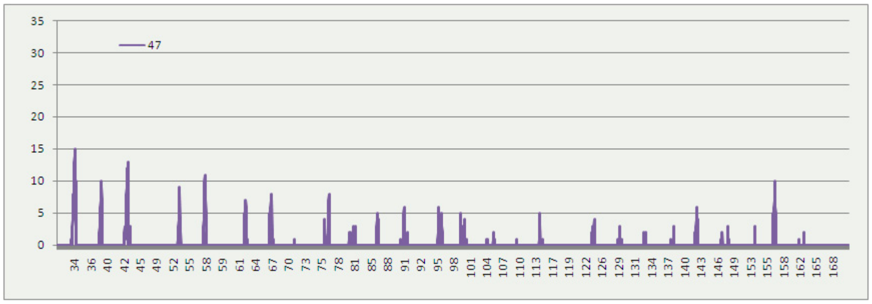

Figure 9: Queue length in node 47 at 40 pkts/s, Single path for MPOLSR, modified

result in more changes in the neighborhood set and the MPR Selector set, and more TC messages will be sent out.

The more TC messages in the network will not be a problem when data rate is low. But it will result in more congestion in the high data rate scenario. In the simulation we did for the single path Ori-MPOLSR, after changing the hold time from 4s to 6s, the delay is reduced from 0.158285s to 0.1019975s.

6 The impacts of TC message

Another modification we did for the MPOLSR before is that we make TC message include all the link between the local node and neighbors, instead of MPR, which is the way used in OLSR, to get more information about the network topology. However, it will also increase the size and the number of the TC message.

Based on the result of Section 5, if we have the TC message just include MPR, as the OLSR do, we can reduce the delay from 0.1019975s to 0.03304282s. The length of the queue at the source node is shown in figure 9.

7 The impacts of routes computation

To study the time used in the Multi Dijkstra algorithm for the routes computation, we prefix the source route 47->96->62->88->80->37 for the data packet in the routing protocol, instead of calculating it. The delay is 0.0192562s, compared with 0.03304282s in section 6. The length of the queue at the source node is presented in figure 10.

As we can see from the delay and the queue length from the figure, the protocol without computation do have better performance. But it is also worth to notice that:

- The computation, which happens in the routing layer, can not be located in the trace file of NS2. According the trace file of NS2, the routing layer

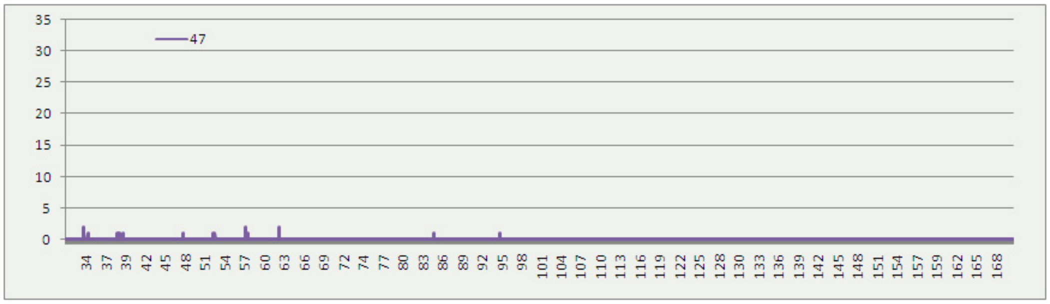

Figure 10: Queue length in node 47 at 40 pkts/s, fixed route

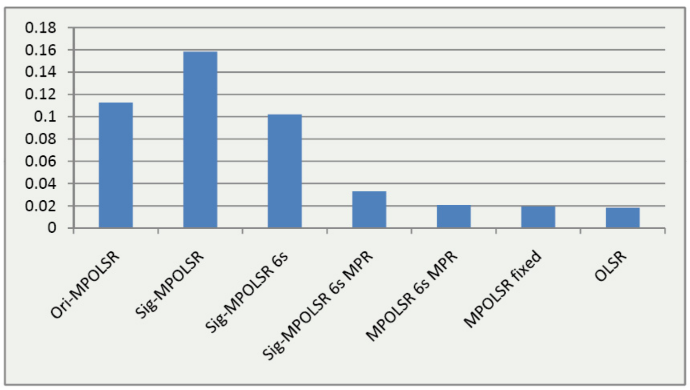

Figure 11: End-to-end delay

send the packets to the MAC layer as soon as it received the data packets, which means the computation still don't take any time. So which step actually result in the delay is hard to locate in NS2.

- Although the congestion at the source node is much better for the fixed route simulation, it has more congestion in the second and third hop nodes. The figure is not shown here because of the space.

8 Summary

This work studies the behavior of outgoing queue of ad hoc protocol OLSR and MPOLSR in NS2. Figure 11shows the delay of the discussed protocols and configurations, which can roughly present the situation of the queue in the networks.

And we can conclude that:

- The congestion of the queue mainly happens at the first or second hop of the path, and last for one or two seconds.

- Decrease the neighborhood hold time will result in more routing messages, and also more congestion.
- Making the TC message includes link with all the neighbors will have longer delay.
- Compared with single path protocol, the multipath protocol in the same situation have better performance and load balancing.
- The computation of the routes in MPOLSR seems to cause longer time, but still hard to locate the problem in NS2.