"An E-2-E Communication Based MPOLSR Protocol for Loop Detection and Isolation in MANET"

Hameed Khan, Ashok Verma, Ajay Lala, Ashish Chourasia
Computer Science & Engineering, GGITS Jabalpur

Abstract

Ad hoc networks consist of a collection of wireless mobile nodes which dynamically exchange data without reliance on any fixed based station or a wired backbone network. They are by definition self-organized. The frequent topological changes make multi-hops routing a crucial issue for these networks. In this paper, we propose a multipath routing protocol named Multipath Optimized Link State Routing (MP-OLSR). It is a multipath extension of OLSR, and can be regarded as a hybrid routing scheme because it combines the proactive nature of *topology sensing* and reactive nature of *multipath computation*. The auxiliary functions as *route recovery* and *loop detection* are introduced to improve the performance of the network.

Keywords: Mobile ad hoc network (MANET), multiple paths, OLSR, MP-OLSR,

INTRODUCTION

A MANET [1][2]is a collection of nodes where the nodes will self configure and self organize themselves forming a wireless medium without any requirement of stationary infrastructure like base station. In these networks each node will not only act as a host but also acts as a router. Due to mobility of nodes, the topology of the network is dynamic that is, it changes most of the time. Some examples where the possible use of Ad-hoc networks are in military, in emergency situation like hurricanes, earth quakes, conferences etc. One of the main issues in Ad-hoc networks is to develop a routing protocol which must be capable of handling very large number of nodes with limited bandwidth and power availability. Also they should respond quickly to the hosts that broken or newly formed in various locations. Many protocols have been proposed to solve these problems in the ad-hoc networks.

In this part of the thesis, we expose our main contribution in the routing protocol: Multipath Optimized Link State Routing (MP-OLSR). It is a multipath extension of OLSR, which can be regarded as a hybrid routing scheme because it combines the proactive nature of topology sensing and reactive nature of route computation.

We probe the multipath routing protocol from design to simulation, and finally the real implementation to validate the functions of the protocol.

Related work

The objective of this work is to evaluate two protocol pro-active / table driven routing protocols namely, Optimized Link State Routing Protocol and Multipath Optimized Link State Routing Protocol. In a proactive routing protocol, called Multipath Optimized Link State Routing for MANET is proposed. The protocol inherits the stability of the link state algorithm. Due to its proactive nature, it has an advantage of having the routes immediately available when needed. MPOLSR is an optimization of a pure link state protocol for MANET. This evaluation is to be carried out through exhaustive literature review and simulation. Then we present the functionality of MP-OLSR. The detailed specifications for the multipath routing are defined. MP-OLSR inherits the topology sensing mechanism from OLSR, which helps the nodes in the network to explore the network topology. The Multipath Dijkstra Algorithm is proposed to obtain multiple paths from the source to the destination. Source routing is employed to forward the packets. To avoid route failure and possible transient loops in the network, Route Recovery and Loop Detection are introduced to improve the performance of the network. The link metric based on queue length information is discussed as a possible replacement of the hop count metric. And in the end of the chapter, the problem of compatibility with OLSR is also illustrated.

Specification of MP-OLSR

Multipath Optimized Link State Routing (MP-OLSR)

The Multipath Optimized Link State Routing (MP-OLSR) can be regarded as a hybrid multipath routing protocol. It sends out HELLO messages and TC messages periodically to be aware of the network topology, just like OLSR. The difference is that MP-OLSR does not always keep a routing table to all the possible destinations. It only calculates the routes when there are data packets need to be sent out. The core functioning of MP-OLSR has two main parts: topology sensing and route computation. The topology sensing makes the nodes get to the topology information of the network, which includes link sensing, neighbor detection and topology discovery. This part gets benefit from MPRs as well as OLSR. By sending the routing control messages proactively, the node could be aware of the topology of the network: its neighbors, 2-hop neighbors and other links. The routing computation uses the Multipath Dijkstra Algorithm to populate the multiple paths based on the information get from the topology sensing. The source route (the hops from the source to the destination) will be saved in the header of the data packets. The medium hops just read the packet head and forward the packet to the next hop. The topology sensing and route computation make it possible to find multiple paths from source to destination. In the specification of the algorithm, the paths will be available and loop-free. However, in practice, the situation will be much more complicated due to the change of the topology and the instability of the wireless medium. So route recovery and loop detection are also proposed as auxiliary functionalities to improve the performance of the protocol. The route recovery can effectively reduce the packet loss, and the loop detection can be used to avoid potential loops in the network.

Topology Sensing

To get the topology information of the network, the nodes use the topology sensing which includes link sensing, neighbor detection and topology discovery, just like OLSR. Link sensing populates the local link information base (Link Set). It is exclusively concerned with OLSR interface addresses and the ability to exchange packets between such OLSR interfaces. Neighbor detection populates the neighborhood information base (Neighbor Set and 2-hop Neighbor Set) and concerns itself with nodes and their main addresses. Both link sensing and neighbor detection is based on the periodic exchange of HELLO messages.

Topology Discovery generates the information base which concerns the nodes which are more than two hops away (Topology Set). It is based on the flooding of the TC messages (optimized by selecting the MPR set). Through topology sensing, each node in the network can get sufficient information of the topology to enable routing. The link state protocol tries to keep the link information of the whole network as mentioned above. By default, the path quality is measured by the number of hops. For the purpose of making the thesis self-contained, this part summarized the Topology Sensing functionality.

Topology Control Message TC messages are broadcasted by each node to the whole network to build the intra-forwarding database needed for routing packets. The format of TC message allows the standard [4] and is defined in [5]. A TC message is sent by a node in the network to declare a set of links, which must include at least the links to all nodes of its MPR Selector set, i.e., the neighbors which have selected the sender node as a MPR. 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. With the broadcast of TC messages to the whole network, the node is able to get the topology information that is more than two hops away.

Link Sensing and Neighbor Detection

The link sensing and neighbor detection are based on the transmission of HELLO messages. Based on the received messages, the procedures called link sensing and neighbor detection are performed to build the link set and 2-hop set. On receiving a packet, the node examines the packet header and each of the message headers. If the message type is known to the node, the message is processed locally according to the specification for that message type. The message is also independently evaluated for forwarding. If parsing fails at any point the relevant entity (packet or message) must be silently discarded. So the incoming packets (TC or HELLO) can be appropriately distributed for next step of process. When receiving a HELLO message, a router must update its Link set for the MANET interface on which the HELLO message is received, and update its Neighbor Set. The algorithm will first find all Neighbor Tuples (henceforth matching Neighbor Tuples) where the message's N_neighbor_addr_list contains any network address which overlaps with any address in the node's Neighbor Address List. If there are no matching neighbor tuples, a new neighbor tuple will be created. If there are one or more matches, then related information has to be updated. the procedure to be performed for Neighbor Set updating. The 1-hop neighbors are then maintained properly based on the exchange of HELLO messages. In addition to the Neighbor Set, the 2-Hop Set also need to be updated. The procedure to be performed for 2-Hop set updating. So the information of the 2-hop nodes is saved in the 2-Hop Set.

Topology Discovery

Link Sensing and Neighbor Detection make the node be aware of its 1-Hop neighbors and 2-Hop neighbors by sending HELLO messages. To get the topology information located more than 2 hops away, Topology Discovery is needed. It is based on the broadcast of TC messages. A node with one or more OLSRv2 interfaces and with a non-empty neighbor set must generate TC messages. A node with an empty neighbor set should also generate "empty" TC messages for a period "hold" time after it last generated a non- empty TC message. Complete TC messages are generated and transmitted periodically on all OLSRv2 interfaces, with a default interval between two consecutive TC transmissions. In addition to the periodic broadcasting, it can be generated in response to a change of contents. Only

MPR can forward the TC messages to the next hop. When receiving a TC message, it is processed according to its type. The node first checks the message is from itself or unavailable. If so, the message must be discarded. Otherwise, the node will populate the related information base set (Advertising Remote Node Set, Topology Set, etc.) based on the received message. So based on the broadcasting and processing of TC messages, the topology information that more than two hops always can be saved in the Topology Set.

Route Recovery

Route Failure in MP-OLSR

By using the scheme of the Topology Sensing, we can obtain the topology information of the network with the exchange of HELLO and TC messages. All this information is saved in the topology information base of the local node: link set, neighbor set or topology set. Ideally, the topology information base can be consistent with the real topology of the network. However, in reality, it is hard to achieve, mainly because of the mobility of the ad hoc network. Firstly, for the HELLO and TC messages, there are certain intervals during each message generation (2s for HELLO and 5s for TC by default) [6]. During this period, the topology might change because of the movement of the nodes. Secondly, when the control messages (especially the TC messages) are being transmitted in the network, delay or collision might happen. This will result in the control message being outdated or even lost. Both of the two reasons mentioned above will result in the inconsistency between the real network topology and the node's topology information base. This means that when a node is computing the multiple paths based on the information base, it might use links that do not exist anymore, and cause the route failure. Furthermore, even if the topology information is correct when the route is being constructed at the source node, the topology might change while the packets are being forwarded in the network. And because of the source routing scheme MP-OLSR uses, the source route cannot be adapted to this kind of changes. For the OLSR, the problem is less serious because it uses hop-by-hop routing. Unlike the source routing, whose routes are decided completely at the source, the nodes in OLSR just forward the packets to the next hop. So there is more chance for a node in OLSR to forward a packet to the next available link.

Route Recovery Algorithm

Several techniques already exist in the literature to deal with the route failures in source routing. DSR handles route errors using route maintenance, mainly by sending RERR messages, which will of G makes the link from E to G unavailable.

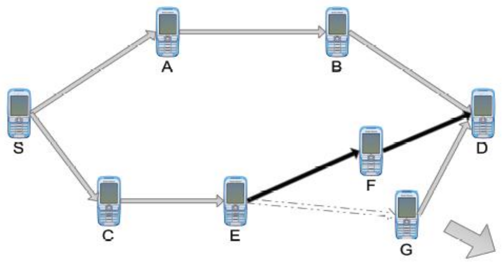

Figure 1.1: An example of route recovery. S is the source and D is the destination.

The movement increases the end-to-end delay significantly. In[8], the authors propose another method to avoid the effect of short term link deterioration by using opportunistic paths in mesh networks. To overcome the disadvantage of the source routing, we propose Route Recovery for MP-OLSR.[7] The principal is very simple: before an intermediate node tries to forward a packet to the next hop according to the source route, the node first checks if the next hop in the source route is one of its neighbors (by checking the neighbor set). If yes, the packet is forwarded normally. If no, then it is possible that the "next hop" is not available anymore. Then the node will recomputed the route and forward the packet by using the new route. We present an example of route recovery. Node S is trying to send packets to D. The original multiple paths we got are S_A_B_D and S_C_E_G_D. However, node G moves out of the transmission range of node E and makes the second path unavailable. The source node S is not able to detect the link failure immediately (because of the delay and long interval of TC messages) and keeps sending the packets along the path and all these packets are dropped during this period if only the source routing is used. With Route Recovery, when the packet arrives, node E will first check if node G is still one of its neighbors, before forwarding the packet according to the source route. If not, node E will recomputed the route to node D, and get E_F_D. Then the following packets will be sent through the new path. Because the Route Recovery just checks the topology information saved in the local node, it will not introduce much extra delay. And most importantly, it will effectively improve the packet delivery ratio of the network.

Loop Detection

Loops in OLSR and MP-OLSR

It is important to mention the LLN (Link Layer Notification) before coming to the problem of the loops of the protocol. LLN is an extended functionality defined in [8], and implemented in different OLSR or MPOLSR simulations and implementations. If link layer information describing connectivity to neighbor nodes is available (i.e. loss of connectivity though absence of a link layer acknowledgement), this information can be used in addition to the information from the HELLO-message to maintain the neighbor information base and the MPR selector of the information bases in node A and B. One transient loop is formed between A and B set.

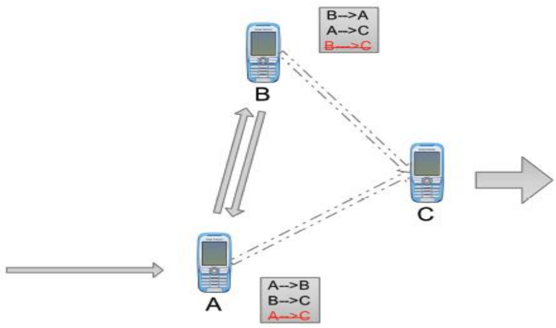

Figure 1.2: An example of loop in the network. The movement of node C results in inconsistency

The routing protocol can act on the acknowledgement from LLN (mainly the loss of links), and remove the corresponding links from its information base. The results of the real OLSRv2 testbed [9] and our work based on NS2 simulation in can show that LLN is very important and effectively improves the packet delivery ratio of the OLSR and MP-OLSR protocol. In theory, the paths generated by the Dijkstra algorithm in MP-OLSR are loop-free. However, in reality, the LLN and Route Recovery which are used to adapt to the topology changes make the loops possible in the network. With LLN, when a node tries to send a packet over a link but fail in the end, the link layer will give a feedback to the routing protocol to notify the link loss. This kind of abrupt interruption will result in additional operations on the topology information base rather than just regular HELLO and TC messages. This means that other nodes cannot be aware of these changes immediately. So LLN might cause some inconsistency of the topology information in different nodes. And with Route Recovery, which might change the path in intermediate nodes, loops can occur temporarily in the network. In Figure 1.1 we give an example of how a loop is generated in the network. Node A is an intermediate node of a path. The packets with source route A_C arrive at node A and need to be forwarded to node C. Then node C moves out of the transmission range of node A and node B,and makes the links A_C, B_C not available anymore. When the new packets arrive at node A, the transmission to node C will fail. Then in node A, the routing protocol will be acknowledged by LLN, and it will remove the link A_C from node A's link set. For node A, although it can detect the link failure of A_C by LLN, it is hard to know the failure of B_C immediately. This is because link B_C can only be removed when the NEIGHB HOLD TIME (6 seconds by default) expires. In the meantime, Route Recovery will be awaken. A new path A_B_C will be established and the following packets will be forwarded along the new path. Then the packets will be redirected to node B. The same operation will be performed in node B: LLN of the failure of B_C, and Route Recovery. Unfortunately, because node B cannot detect the link failure of A_C immediately, and the new path obtained by Route Recovery is B_A_C. Thus the packet will be returned to node A, and node A to B again, creating a loop. This is not a permanent loop, but a transient loop which will exist for several seconds and will disappear when the related link expires. However, this kind of temporary loops will block the links in the loop and congest the related transmission area. In [9], the authors also address the looping issues in OLSRv2, and LLN will significantly increase the number of loops.

Source Route Loop Detection

In the authors introduce two types of loop detection techniques: LD-Mid (Mid-Loop Detection) and LD-Post (Post-Loop Detection). LD-Mid just compares the address of the next hop against the address of the previous hop, so it is only able to detect "two-way" loops between 2 nodes. LD-Post records all incoming packets that need to be forwarded and compares against each new incoming packet to see if the same packet has traversed this node before. So it can detect loops that are farther away, by taking more memory. When a loop is detected, the Packet Discard strategy is used to drop the packets that are unlikely to reach the destination but only increase the load of the network. For MP-OLSR, we propose a simple method based on source routing that can effectively detect loops without causing extra cost of memory: after the Route Recovery is performed, we can get a new set of multiple paths from the current node to the destination. The node will compare the first new path with the ancient source route in the packet. We can verify if the new path includes the nodes that the packet had crossed before. If the answer is no, it means that there will be no loop in the future, and we will make use of the new path. If the new path includes the node that the packet have passed before, there is high probability that a loop will happen (a very rare case is that the failed link is recovered in this short period, in several milliseconds, then the loop is released). MP-OLSR will switch to the next path of the multiple paths set, until all the paths have been verified. If there is no suitable path, the packet will be discarded. For the example in Figure 1.1, node A will get a path A_B_C by Route Recovery. Then when the packet arrives at node B, a new path B_A_C will be generated because of link breakage of B_C. Node B will compare the new one with the ancient source route A_B_C in the packet. We will find that the packet has already crossed node A, and so there might be a loop. Then we will try to find if there is any other possible path, or else the packet will be discarded. Compared with LD-Post, which needs to keep a record of all the incoming packets, our loop detection mechanism could effectively detect the possible loops in the network without consuming extra memory space. By reducing the loops in the network, the network congestion can be reduced. So the performance of the network can be improved, especially the end-to-end delay.

RESULTS AND DISCUSSIONS

Packet Delivery Fraction

The figure shows that initially for 2 nodes the PDF is highest for MPOLSR protocol while the PDF for OLSR are moderate. As the no of nodes increases the PDF for MPOLSR and OLSR decreases gradually but again MPOLSR is greater than OLSR protocol for 10 nodes. The PDF is highest when increase 20 nodes we found both protocol approximately equal and finally 30 node case both protocol decrease gradually with minor difference.

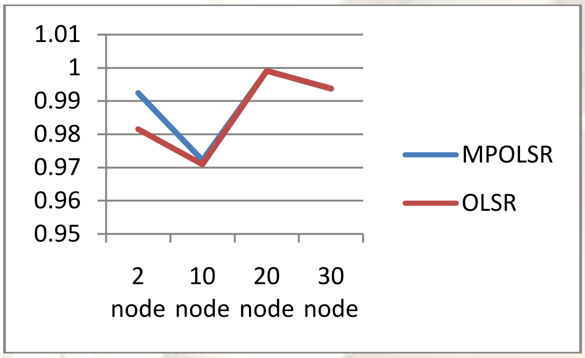

Fig 7.1: Packet Delivery Fraction

Throughput

The figure shows that initially for 2 nodes the THROUGHPUT is highest for OLSR protocol while the THROUGHPUT for MPOLSR is moderate. As the 10 nodes increases the THROUGHPUT for MPOLSR is greater and OLSR decreased gradually but again MPOLSR is greater than OLSR protocol just approximate for 20 nodes. The THROUGHPUT is approximately equal when we set 30 nodes OLSR and MPOLSR protocol in network.

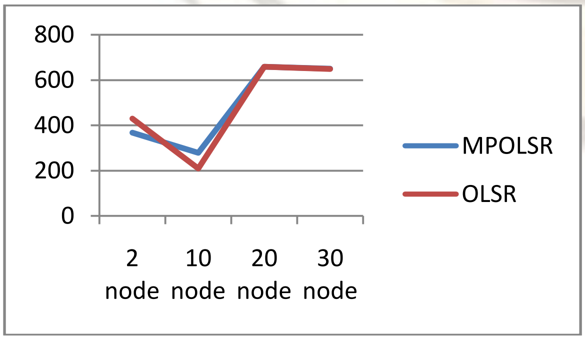

Fig 7.2: Throughput

End to End Delay

The figure shows that initially for 2 nodes the End-2- End delay is minor difference between OLSR and MPOLSR protocol. As the 10 nodes increases the End-2- End delay for OLSR is greater as compared to MPOLSR. But again MPOLSR and OLSR protocol just approximate equal for 20 nodes. The End-2- End delay is gradually increase when we set 30 nodes MPOLSR and OLSR protocol are also increase but gradually lesser then MPOLSR in N/W.

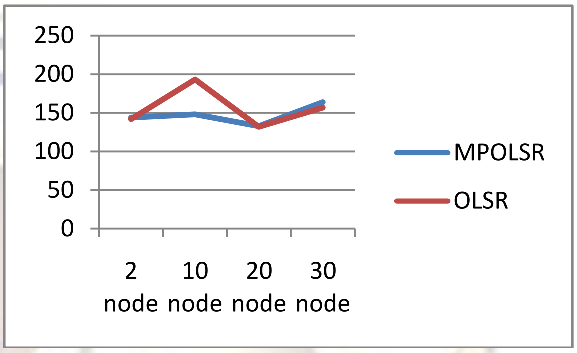

Fig 7.3: End-2- End delay

Conclusion

In this chapter, the specifications of MP-OLSR are introduced. The topology sensing (based on OLSR) and route computation are basic procedures for multipath routing. To improve the performance of the protocol, the route recovery and loop detection are proposed to avoid route failure and reduce transient loops in the network. We proposed a queue length metric to evaluate the link quality. The queue length information is saved in TLVs and propagated to the whole network by HELLO and TC messages. The compatibility between MP-OLSR and OLSR are also studied to make the single path routing and multipath routing is able to cooperate with each other. This further study highlights the interest of multiple paths routing to improve quality of experience over self-organized networks.

References

[1] A. A. Pirzada, C. McDonald, and A. Datta, “Performance Comparison of Trust-Based Reactive Routing Protocols,” IEEE Transactions on Mobile Computing Vol. 5, No.6 ,pp. 695- 710, 2006 .

[2] Mamoun H. Mamoun “ Important Characteristic of Differences between DSR and AODV Routing Protocol” 2007

[3] Clausen, T. & Jacquet, P. (2003). IETF Request for Comments: 3626, Optimized Link State Routing Protocol OLSR.

[4] Clausen, T., Dearlove, C., Dean, J., & Adjih, C. (2009c). IETF Request for Comments: 5444, Generalized Mobile Ad Hoc Network (MANET) Packet/Message Format.

[5] Clausen, T., Dearlove, C., & Jacquet, P. (2009b). IETF Internet Draft, The Optimized Link State Routing Protocol version 2, draft-ietf-manet-olsrv2-10.

[6] Jiazi Yi, Eddy Cizeron, Salima Hamma, Benoît Parrein and Pascal Lesage*Implementation of multipath and Multiple Description Coding in OLSR This is a proceeding of 4th OLSR Introp/Workshop, Ottawa, Canada.

[7] Jiazi YI, Asmaa ADNANE, Sylvain DAVID, Benoˆıt PARREIN Multipath Optimized Link State Routing for Mobile ad hoc Networks *Preprint submitted to Ad Hoc Networks September 28, 2010*

[8] Speakman, L., Owada, Y., & Mase, K. (2008). An analysis of loop formation in OLSRv2 in adhoc networks and limiting its negative impact. In IEEE International CQR Workshop, Naples, Florida, USA.

[9] Clausen, T. & Jacquet, P. (2003). IETF Request for Comments: 3626, Optimized Link State Routing Protocol OLSR.