Label Routing Protocol: A New Cross-layer Protocol for Multi-hop Ad hoc Wireless Networks

Yu Wang
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431
ywang13@fau.edu

Jie Wu
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431
jie@cse.fau.edu

***Abstract*— Compared to the traditional wireless network, the multi-hop ad hoc wireless network is self-configurable, dynamic, and distributed. During the past few years, many routing protocols have been proposed for this particular network environment. While in wired and optical networks, multi-protocol label switching (MPLS) has clearly shown its advantages in routing and switching such as flexibility, high efficiency, scalability, and low cost. However MPLS is complex and does not consider the mobility issue for wireless networks, especially for ad hoc wireless networks. This paper migrates the label concept into the ad hoc network and provides a framework for the efficient label routing protocol(LRP) in such a network. The MAC layer is also optimized with LRP for shorter delay, power saving, and higher efficiency. The simulation results show that the delay is improved significantly with this cross-layer routing protocol.**

***Index Terms*— ad hoc networks, label routing, MPLS, path, routing protocol, wireless networks.**

I. INTRODUCTION

A router in a wired network typically requires multiple network interfaces to act as a router or a forwarding node. In multi-hop ad hoc wireless networks (or simply ad hoc networks), any node with a wireless network interface card can operate as a router or a forwarding node, since it can receive a packet from a neighboring node, look up a route based on the destination IP address of packets, and then transmit the packet to another neighboring node using the same wireless interface. This characteristic often causes ad hoc networks to be viewed quite differently from traditional networks.

In wired and optical networks, multi-protocol label switching (MPLS) has already shown its great advantages with the label concept such as flexibility, high efficiency, and easy integration with other layer-3 protocols (network layer). This paper migrates the label concept into the ad hoc networks and proposes a more traffic-efficient and power-efficient source routing protocol, called Label Routing Protocol (LRP), in ad hoc networks. LRP achieves a virtual connection-oriented protocol in the ad hoc network with QoS, traffic engineering and multicast capability. Furthermore, as a cross-layer routing protocol, the MAC layer is also optimized for shorter delay, power saving, and higher efficiency.

LRP adopts the label-switching paradigm from the MPLS and ATM networks. Therefore it inherits many advantages from them such as simplicity, efficiency, and flexibility. In the mean time the consumed energy could be smaller than other routing protocols in ad hoc networks such as DSDV, AODV, DSR, ZRP, etc.

Since traffic engineering is more and more important in the real network environment, ad hoc networks should provide some traffic engineering features excluding QoS such as connection admission control, traffic policing or shaping service, traffic prioritization, etc. In addition, some paths or connections can be ranked for quick rerouting and backup paths may be pre-provisioned for rapid restoration in order to support critical services. Sometimes network operators need routing flexibility such as policy-based routing or QoS-based routing to provide diverse services for customized services. But it is very difficult to approach them with the traditional distributed routing protocol, especially in ad hoc networks. With the label concept, all of these features could be easier to achieve with LRP, which also supports virtual connection oriented and source routing solutions.

Another advantage of this protocol is that nodes can discover and maintain the path by label instead of IP address and complicated routing algorithms. All intermediate nodes in the virtual connection or path can forward packets more efficiently with this protocol combined with an optimized MAC layer protocol.

The entire architecture of LRP was developed in the summer of 2004. The fast forwarding scheme, which is based on shortcuts in the layer below IP layer, is similar to the idea in Dr. Ramanathan's new architecture for ad hoc networks [14]. The main ideas of his paper are the cut-through solution in the physical layer and forwarding method which is similar to our proposed scheme. But with years of experience in the industry, we think LRP, which supports fast forwarding processing in the MAC layer, is better and more operable because the physical layer always focuses on the processing procedures related to radio signal, such as channel coder/decoder, scramble/descramble, modulation/demodulation, spreading/despreading and so on. Also, we refrain from changing the IP layer as much because there are a lot of well-developed software and hardware modules in the industry and they can be modified slightly to support LRP.

The rest of this paper is organized as follows. In section II, we provide some preliminaries of this new routing protocol. Section III describes LRP in detail. Some optimizations for this new protocol on the MAC layer are introduced in section IV. Section V presents our scheme, some models and results of our simulation. The last section concludes this paper.

II. PRELIMINARIES

A. Overview of Routing Protocols

The limited resources of ad hoc networks have made designing an efficient and reliable routing strategy a very challenging issue. So far, many routing protocols have been proposed for ad hoc networks. These protocols can be classified into three different groups: proactive, reactive, and hybrid [1].

In proactive routing protocols, the routes to all destinations (or parts of the network) are determined at start up, and are maintained by using a periodic route update process. DSDV (destination-sequenced distance vector) [9], WRP (wireless routing protocol) [8], and GSR (global state routing) [6] are proactive routing protocols.

Reactive protocols are on-demand routing protocols that were designed to reduce the overhead of proactive protocols by maintaining information for active routes only. In other words, routes are determined when they are required by the source using a route discovery process. A number of different reactive routing protocols have been proposed such as DSR (dynamic source routing) [7], AODV (ad hoc on-demand distance vector) [5], TORA (temporally order routing algorithm), etc. DSR is a source routing reactive protocol. In DSR, each data packet carries the complete information from source to destination. Each intermediate node forwards these packets according to the information in the header of each packet.

AODV (ad hoc on-demand distance vector) is based on DSDV and DSR. It uses periodic hello messages, the sequence number of DSDV and a route discovery procedure similar to the procedure in DSR. However there are two major differences between DSR and AODV. One is that AODV does not need to keep complete route information in each data packet. The other is that route replies in AODV only carry the sequence number and IP address of the destination node.

Hybrid routing protocols are both proactive and reactive in nature. These protocols are designed to increase scalability by allowing nodes with close proximity to work together as a backbone structure to reduce the route discovery overhead. Most hybrid protocols proposed are zone-based, which means that the network is partitioned into a number of zones by each node. ZRP (zone routing protocol) [10], DST (distributed spanning trees based routing protocol) [11] and DDR (distributed dynamic routing) [12] are typical hybrid routing protocols.

B. Overview of MPLS

Currently the IP network uses destination-based forwarding to determine the next hop of a packet. Route lookup is based on the destination IP address. The longest-prefix match required in IP address lookup was traditionally implemented in software and viewed as too slow for core networks. Although recent advances in IP address lookup techniques and hardware implementations have allowed destination-based packet forwarding to perform at a higher speed, lookup techniques based on simpler forwarding information provide certain advantages as embodied in multi-protocol label switching (MPLS) [2].

The label-switching paradigm in MPLS and ATM performs lookup based on a short fixed-length label to determine the next hop for each cell. Label switching uses a label to directly index into a connection table entry to determine the next hop, lending itself to simple lookup implementation with a high forwarding rate. Label-switching forwarding is also considered more attractive than destination-based forwarding because of its flexibility. If the path is determined only by the destination address of a packet with the same destination, it can not follow different service classes or rates. On the other hand, multiple paths with the same destination can be setup for different service classes or rates with label-switching forwarding. Thus label switching has been considered a better candidate for traffic engineering than the traditional IP paradigm.

III. LABEL ROUTING PROTOCOL (LRP)

Since most nodes in ad hoc networks have limited energy and processor resources, protocols and algorithms running on them should have a high level of efficiency and a low level of complexity. Although MPLS is well designed for wired networks, MPLS does not consider the mobility issue for wireless networks, especially for ad hoc wireless networks. It is also too complex to be implemented in an ad hoc network. But the label concept and label-switching paradigm can be migrated into ad hoc networks. This is how we designed the new routing protocol, label routing protocol (LRP).

Similar to MPLS, LRP is based on the label concept. It is a virtual connection-oriented protocol that is able to setup, configure, and maintain a path between two or more endpoints.

The path from source node to destination node works as a tunnel identified by multiple labels and located between layer 2 and layer 3. We could name it a layer-2.5 tunnel. This layer-2.5 tunnel is used to simulate a permanent virtual connection for purposes of efficiency and QoS. It is a low-overhead virtual circuit solution for the connectionless oriented network. With this layer-2.5 tunnel, intermediate nodes can provide fast and efficient forwarding without checking the IP address and accessing a large routing table in the memory of the host CPU.

This section describes LRP on ad hoc networks, including the definition of label, label routing tables, path discovery, path reservation, label forwarding, and path maintenance. Furthermore the optimized MAC layer for LRP is presented in the next section.

TABLE I
STRUCTURE OF LABEL

Multicast Flag	Label	COS	TTL

A. Label

A label is a short, fixed-length identifier. Multiple labels can identify a path or connection from source node to destination node. The structure of the label is shown in Table I.

The first part of the label is related to multicast connection. It contains two parts, the first bit is a multicast flag and the following 3 bits are the number of total connections that come from this node. If the value of the flag is one, it means this is a multicast connection. The maximum number of total connections is eight. If the value of the flag is zero, it means this is a unicast connection and following 3 bits are 001, which means only one connection comes from this node.

Label is a 20-bit field after the multicast flag, which includes a node identifier; therefore all labels are unique in the network. Since we are focusing on a wireless network, the coverage of each node could be different because of the power of the radio transmission module and obstacles around these nodes. If a label could be shared between non-adjacent nodes, it could make a conflict after node movement or radio power adjustment. Also if nodes move quickly, non-adjacent nodes could be adjacent nodes very soon. One label conflict could result in a connection interruption in this scenario. In order to prevent the connection from interrupting, we must make the label unique in our ad hoc networks.

After label field there are 3 bits for COS, which means class of service. Here we have 8 classes of service, from level 0 to level 7. Each level corresponds to one queue. Level 7 is the highest priority level queue and level 0 is the lowest priority level. For the purpose of more efficient processing, all queues located in the baseband module of the wireless interface subsystem, normally in the baseband IC or extended SRAM of the baseband IC. Queues are processed in strict priority order until all queues are empty.

The last field of this label is TTL (time-to-live). All label information has a time-related restriction. Once the time is out for a label, all corresponding entries will be deleted from label routing table. Therefore we can maintain the label routing table and keep it fresh without making it too big.

B. Label routing table

The main idea of this routing protocol is based on the concept of label, which combines layer 2 and layer 3 together for the purpose of highly efficient routing in the ad hoc network. Instead of IP routing table we create a new table, the label routing table, to implement routing, packet forwarding, and path management. This label routing table is composed of two parts. The first of its two parts is located in the memory of host CPU and is named label routing table for easy understanding; the other smaller one, label cache table, is located in the baseband module of the wireless interface subsystem. The reason to separate it into two parts is explained in section IV.

Each entry of the label routing has the following fields:

- $Label\_in$ is the field for the label of the packet which should be processed.
- $Label\_out$ is the field for the label of the packet which should be forwarded. $Label\_out$ field is bound with $Label\_in$ field to identify one path or connection. MAC address of the node of the next hop could be embedded in $Label\_out$ field to provide higher efficiency. Also this field could have multiple labels for multicast connection. Zero value of this field represents that the destination node of this path is reached and this packet will be transmitted to the host CPU for further processing.
- Service level indicates the COS value in the label for QoS service. All service levels of sub-connections in the multicast traffic have the same value, in other words, they are in the same service level.
- Source field is the source node ID of this path or connection.
- Destination field is the destination node ID of this path or connection
- Destination sequence number is the sequence number of destination node
- Lifetime is the time-to-live of this path

Each entry of the label cache table has the $Label\_in$ field and $Label\_out$ field. They are all copied from the label routing table by host CPU. In other words, they are controlled by host CPU and can be deleted anytime. If $Label\_out$ of a receiving packet is zero, this packet will be transmitted to host CPU for further processing. The MAC address of next node in the path could also be placed in label-out field if the current node isn't the destination.

C. Path Discovery

With LRP, communication between end points relies on a virtual tunnel. In order to set up this tunnel, a path or connection must be established between the source and destination nodes and identified by labels. If there is no path which can reach the the destination node in its local label routing table, the source node will initiate a path discovery. This phase is similar to the route discovery of AODV [5]. But we integrate the label concept, QoS, and multicast capability into this phase.

To discover a path to the destination node, the source node creates a label request ($LREQ$) message. This packet contains IDs of the source node and destination node, sequence number of the source and destination nodes, and service level required. The $LREQ$ also contains broadcast ID and a hop count that is initialized to zero.

All nodes that receive this message will increment the hop count value. If a node does not have any information about the destination node, it will record the neighbor's ID where the first copy of $LREQ$ is from and send this $LREQ$ to its neighbors. $LREQs$ from the same node with the same broadcast ID will not be processed more than once.

Figure 1 gives an example of an ad hoc network. In this example, there are eight nodes with nine duplex connections links and one uni-connection link. A solid line between two

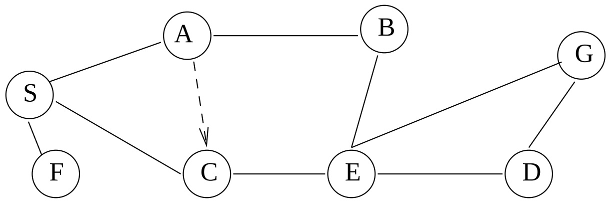

Fig. 1. Network topology example

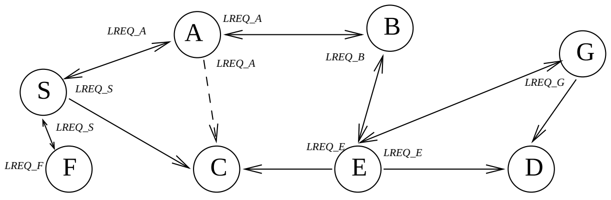

Fig. 2. Sending label request ($LREQ$)

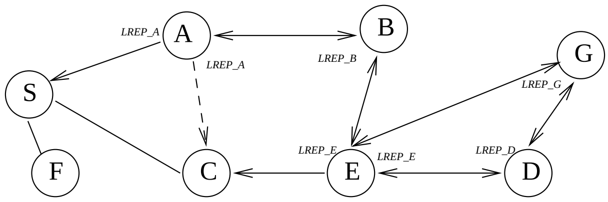

Fig. 3. Sending label reply ($LREP$)

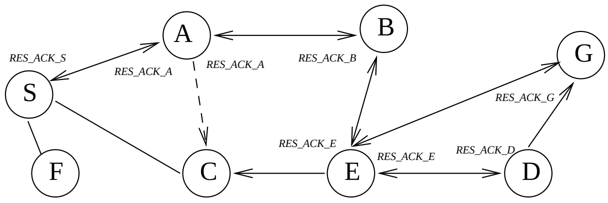

Fig. 4. Reserve acknowledgment ($RES-ACK$)

nodes represents a duplex connection link. A dashed line represents a uni-connection link. For instance, the connection between A and C is a uni-connection in Figure 1, while the connection between E and D is a duplex connection.

As a example, node S is going to communicate with node D. First of all, if D is not its neighbor and there is not any related element in its local label routing table, S will create a label request message ($LREQ$) and send it out to all its neighbors. Figure 2 illustrates the propagation of $LREQ$ across the network and the reverse path at every node. The reverse path entry is created for the transmission of the reserved label for this path. This label is embedded in the label reply message ($LREP$). The reserve path entry will be maintained long enough for $LREQ$ to traverse and for some nodes to send a $LREP$ to the source node. If this node is the destination or knows a path to destination, it will check the sequence number of the destination node in the current path in order to avoid old path information. It should be at least as great as the value in the $LREQ$. Otherwise the existing path in the table will be ignored. If $SEQ_n \geq SEQ_{LREQ}$, it will also check if the current queue, which is the service level requested by source node, has enough capability for this new request. If not, it will check the queue with a lower service level. If queue of service level 0 still doesn't have enough capability, this request will be ignored. For instance, in the Figure 2, node C receives $LREQ$ from node S, but it has to ignore and discard this $LREQ$ because its resources do not meet the requirement in the $LREQ$ from S.

If all requests can be satisfied by this node and it does not have knowledge of the destination node, it will increment the hop count in the $LREQ$ by one and then broadcast it to its neighbors. Any duplicated $LREQ$ (same source node ID and same broadcast ID) will be discarded. In Figure 2, node C will ignore the $LREQ$ from A and the $LREQ$ from E because it already received the $LREQ$ from node S with same source node and broadcast ID.

Normally each network has a reasonable maximum hop count. There isn't any path in the network for which the hop count can be more than this specified threshold. In other words, any label request message ($LREQ$) which has a hop count larger than this limitation will be ignored.

D. Path Reserve and Label Forwarding

If hop count, sequence number, or service level can not meet the requirement mentioned above, the $LREQ$ will be ignored and discarded. If the sender of $LREQ$ does not receive a reply message, it will resend this $LREQ$ to account for the possibility that the message was lost. In order to reduce the overhead of network traffic, each node can re-send $LREQ$ only one time for each connection request.

If hop count, sequence number, and service level are all acceptable, the node will create a $LREP$ with the total hop count of this path, the new sequence number of destination node that is the largest one between SEQ_n and SEQ_{LREQ}, the service level matched, and a label from its label pool. Then this $LREP$ will be sent back to the source node along the reverse path entry. Figure 3 shows the propagation of $LREP$ along the reserve paths.

With LRP, the path between source node and destination node is composed of multiple segments. In other words, the path is separated by segments and all data packets are relayed by these segments. Each segment is a real connection between two nodes and labelled by the sending-side node of $LREP$ in this segment. For example in the path S - A - B - E - D (Figure 5), node A sets up the label of the segment between S and A. Nodes B, E, and D set up the labels of the segments between A and B, B and E, and E and D respectively.

Each intermediate node on the path will record the label in the valid $LREP$ message into the $label\_out$ field of its routing table. Also it will randomly select a valid label from its label pool and then create another $LREP$ message with this label and send it out along the reserve path.

In order to avoid uni-directional links in the path, every node that receives $LREP$ must send a RES_ACK back to the sender of that $LREP$ to make sure the current segment is a bi-directional link. Otherwise, the sender will put the target node into its blacklist and ignore all $LREQs$ from

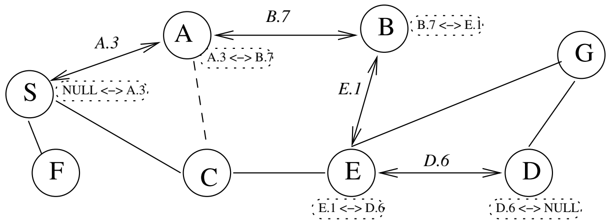

Fig. 5. Label reserved

TABLE II
LABEL ROUTING TABLES FOR THE PATH FROM S TO D

Label Routing Table on the node S

L_in	L_out	Level	Source	Dest.	Dest.Seq#	TTL
NULL	A.3	5	S	D	136	10
A.3	NULL					

Label Routing Table on the node A

L_in	L_out	Level	Source	Dest.	Dest.Seq#	TTL
A.3	B.7	5	S	D	136	10
B.7	A.3					

Label Routing Table on the node B

L_in	L_out	Level	Source	Dest.	Dest.Seq#	TTL
B.7	E.1	5	S	D	136	10
E.1	B.7					

Label Routing Table on the node E

L_in	L_out	Level	Source	Dest.	Dest.Seq#	TTL
E.1	D.6	5	S	D	136	10
D.6	E.1					

Label Routing Table on the node D

L_in	L_out	Level	Source	Dest.	Dest.Seq#	TTL
D.6	NULL	5	S	D	136	10
NULL	D.6					

this node for a while. Message exchange between $LREQ$, $LREP$ and RES_ACK works with a handshake mechanism. For example, in Figure 4 the link between node A and C is a uni-directional link, so node C is in the blacklist of node A if node A has a connection history with node C and cannot receive anything from C.

After RES_ACK related with this $LREP$ is received from the upper link, the node will fill the $Label\_in$ field of the label routing table with the $Label\_in$ the RES_ACK and then create another entry in the label routing table which has the reverse order of the previous label pair. In other words, the previous $Label\_in$ will be copied to the $Label\_out$ field in the second entry and previous $Label\_out$ will be copied to the $Label\_in$ field in the second entry. The reason for making another reverse entry in the table is to help the hardware (baseband module) swap labels of the received valid data packets at the fastest speed.

With these two entries of one path in the routing table, as shown in Table II, duplex communication can be provided instead of one way communication. All intermediate nodes only need to find the available entry indexed by label in the packet, swap it with respective $Label\_out$ of this entry, and then send it out to the next relay node.

If the topology of the network is meshed enough, the source node could receive more than one $LREP$. There is a hop count field in the $LREP$. This field records the total number of hops of the path. The source node will choose the smallest hop count from the $LREP$s in the specified limited time. All $LREP$s that are received after this time threshold will be ignored. And if some available $LREP$s have the same hop count, the path that has largest destination sequence number, which means it is the latest path, will be the final winner.

As a result, the virtual connection (path) is created and identified by multiple labels and ready for communication.

E. Path Maintenance

Compared with wired communication, wireless communication is not stable and its performance could vary dramatically from moment to moment because of mobility and the surrounding environment. The status of a duplex channel could be changed to uni-directional channel or dead and then come back to live again. In order to enhance the robustness of wireless communication, all wireless communication protocols, including LRP, must have the ability to deal with this scenario.

Figure 6 is an example of an established path from node S to node D by LRP. C.2, E.5 and D.1 are labels reserved for this path.

If node E detects some problems on the downstream link to node D, for instance a broken link or one way communication, as shown in Figure 7, node E will send an ERR message to its upstream node C to tell C that the downstream link to D has failed. C will start another path discovery from C to the destination node D and then buffer the data packet received from its upstream node S. After creating a new path from node C to D, C will forward all buffered data packets to the destination node as shown in Figure 8. Source node S does not need to know the modification of this path.

However if node C cannot setup a new path from C to D in the specific time or its buffer overflows, C will send an ERR message to its upstream node as well, and its upstream node S will re-initialize another path discovery phase to implement communication.

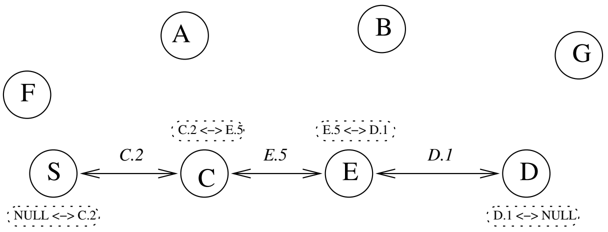

Fig. 6. Sample path

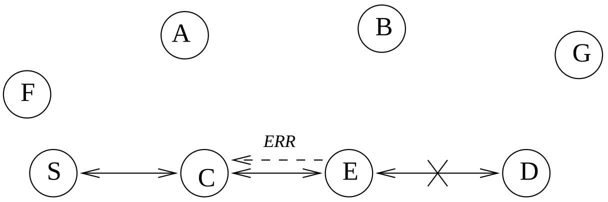

Fig. 7. Path maintenance

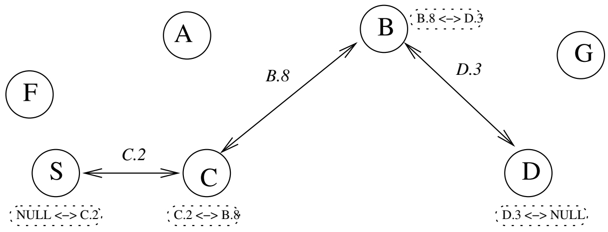

Fig. 8. New path after path maintenance

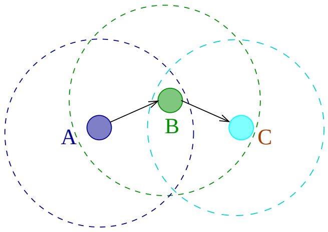

Fig. 9. Forwarding example

IV. MAC LAYER OPTIMIZATION WITH LRP

As we know, energy is a critical issue in MANET. Therefore MAC layer optimization for LRP will focus on this issue. With LRP, the packets are transmitted from the source to the destination node(s) through multiple intermediate nodes on the path. In other words, the core processes on these intermediate nodes are label swapping and packet forwarding.

Since packet forwarding is not a very complicated process, if we can take it out of the job list of the host CPU and have it done by hardware, it could save a lot of resources of the host CPU including consumed energy and processing power. With current electronic technology, a baseband module, including baseband ASIC and extended SRAM, could be dramatically faster if we can find a way that does not need to interrupt the host CPU frequently.

In order to swap labels quickly, we must access the label routing table more efficiently. At the intermediate node, only $Label\_in$ and $Label\_out$ (with MAC address) are necessary for swapping and forwarding. Thus we can copy them to the label cache table in the baseband module of the wireless interface subsystem. The baseband module is responsible for swapping labels and forwarding packets. Also, it will manage queues of 8 service levels for QoS services. The host CPU only needs to access this table when it should be updated according to the original label routing table in the host memory.

With Figure 9 as an example, let's go through the steps for sending a packet from A to C via a forwarding node B with the original DCF mode of IEEE 802.11 MAC layer protocol.

The first step is the upstream channel access by A and B: this follows the standard RTS / CTS / DATA / ACK sequence [4].

The next step is packet processing at node B: it receives the packet on its NIC (network interface card), transfers the packet to the host which looks up the next hop IP and MAC address based on the packet's destination IP address, and transfers the packet back to the NIC again.

TABLE III

PAYLOAD OF ACK/RTS CONTROL PACKET

T		
ACK flag	MAC addr	
RTS flag	MAC addr	Label_out

The third step is downstream channel access by B and C, followed by packet transmission to C. Here there are 2 independent channel accesses which mean double control overhead: the channel has to be sensed free for the DIFS period plus a random time interval twice [3].

In order to eliminate the overhead of multiple channel accesses mentioned above, we can combine the ACK (to the upstream node) with the RTS (to the downstream node) in a single ACK/RTS packet that is sent to the MAC broadcast address [3].

As illustrated in Table III, the payload of the ACK/RTS packet contains the MAC address of the upstream node, the MAC address of the downstream node, and a label intended for use by the downstream node to determine its next hop.

The reservation for the downstream hop is attempted only after successfully receiving the DATA packet from the upstream node. Since the downstream node (and all other neighboring nodes of the forwarding node) is assured to be silent until the completion of the ACK from the forwarding node, contention-free channel access to the downstream transmission can be guaranteed. If it fails when the downstream node fails to respond to the ACK/RTS with CTS, the forwarding node then buffers the packet in the baseband module and sends the original 802.11 RTS to the downstream node according to the original IEEE 802.11 protocol.

V. SIMULATION

In this section, we present some models, schemes and results of the simulation that have been implemented for the proposed LRP. The entire simulation has been implemented with OPNET Release 11.0 [13] using the optimized MAC layer, IP layer with LRP protocol, and a two-way radio propagation model with constant antenna heights. Besides the Label Routing Protocol we created, the AODV module provided by OPNET has been simulated at the same time for the comparison. By default, all protocols use 1-second Hello interval and collect 1-hop information. Since the procedures of path discovery, reserve and maintenance of LRP are similar to AODV, their routing overheads are similar too. Therefore we choose delay, which is the most important factor for the real-time traffic, to compare these two protocols.

The simulation network contains 25 mobile nodes in a 5 X 5 kilometer area. All nodes in the network are configured to move randomly with uniform speed and random direction. The range of uniform speed can be changed for different scenarios. Each node in all these scenarios is designed by the layered model, which is shown in Figure 10. With this scheme, it is easy to migrate other routing protocols into this simulation package. Also Figure 11 describes the MAC layer model, which is optimized for LRP and fast forwarding.

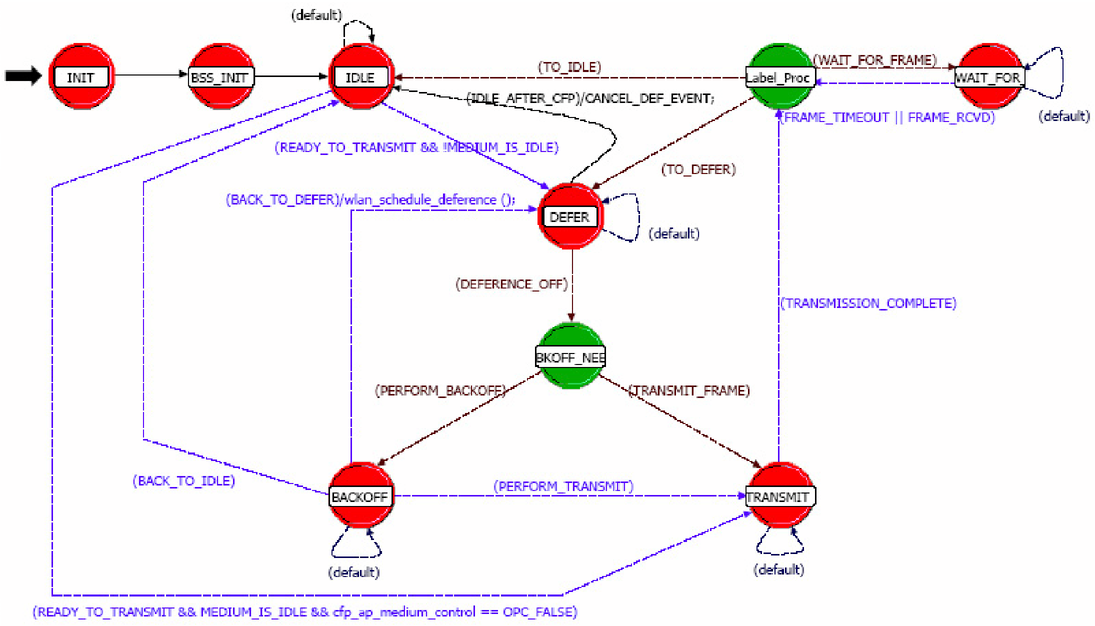

Fig. 11. Optimized MAC layer model

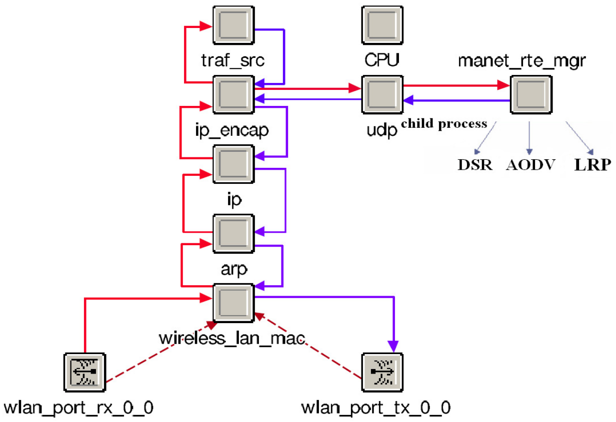

Fig. 10. Mobile node model

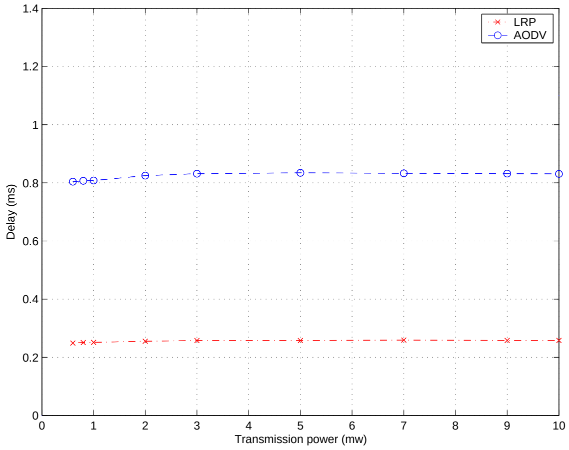

Fig. 12. Simulation result of scenario 1

For the comparison between LRP and AODV, the simulations were implemented in two scenarios, different transmission powers and different movement speeds.

At first we present the comparison on delay between LRP and AODV under a series of configurable transmission powers. The traffic data rate is 120kbps and the speed of random movement of each node is uniform between 0 and 15m/s. The simulation time is 3600 seconds.

Simulation result of this scenario is illustrated in Figure 12. We see that the delay with LRP protocol is less than half of the delay with AODV irrespective of the value of transmission power. Obviously this is the benefit of the shortcut in the MAC layer of all intermediate nodes along the path.

Since we are studying wireless networks, all nodes in the network have capability to move. The second scenario is focused on movement of nodes. We change the movement speed at each step of the simulation, and its value is configured as a uniform value between minimum and maximum speeds we setup. In this scenario, traffic is 120kbps and transmission power of each node is 7mW. The simulation results presented in Figure 13 show that the delay of LRP is still significantly smaller than AODV.

Overall, LRP and optimized MAC layer achieve shorter delay than AODV under selected configurations of the network. In other words, the processing time consumed on the host CPU of each intermediate node is shorter than AODV. It obviously means that the power consumed on these host CPUs is smaller than AODV.

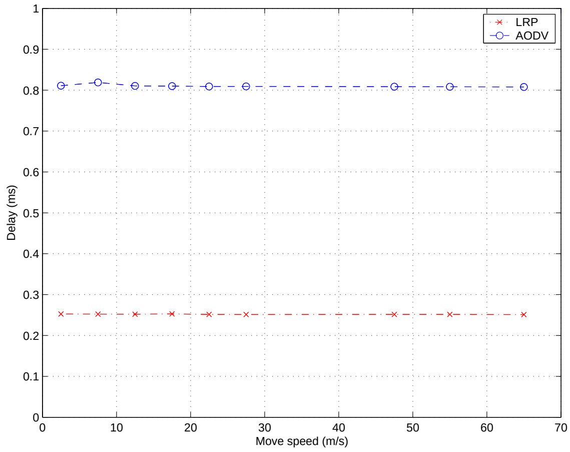

Fig. 13. Simulation result of scenario 2

VI. CONCLUSIONS

In this paper we migrate the label concept into the ad hoc network and provide a framework for a cross-layer and highly efficient routing protocol, Label Routing Protocol. The MAC layer is also optimized with LRP for shorter delay, power saving, and higher efficiency. The simulation results show that the delay is improved significantly with this cross-layer routing protocol.

ACKNOWLEDGMENT

This work was supported in part by NSF grants, ANI 0073736, CCR 0329741, CNS 0422762, CNS 0434533,and EIA 0130806.

REFERENCES

[1] M. Abolhasan, T. Wysocki, E. Dutkiewicz. A review of routing protocols for mobile ad hoc networks. *Ad Hoc Networks*, (2):1-22, 2004.
[2] A. Leon-Garcia, I. Widjaja. *Communication Networks*, pp.727-728. McGraw Hill, 2005
[3] A. Acharya, A. Misra, S. Bansal. A Label-switching Packet Forwarding Architecture for Multi-hop Wireless LANs, WoWMoM, ACM, Atlanta, September, 2002.
[4] IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Computer Society*, 1997.
[5] E.M. Belding-Royer, C.E. Perkins. Evolution and future directions of the ad hoc on-demand distance-vector routing protocol. *Ad Hoc Networks*, (1):125-150, 2003.
[6] T.W.Chen, M.Geria. Global state routing: a new routing scheme for ad-hoc wireless networks. IEEE ICC, 1998.
[7] D.Johnson, D.Maltz, J.Jetcheva. The dynamic source routing protocol for mobile ad hoc networks. *draft-ietf-manet-dsr-07*, Internet Draft 2002.
[8] S. Murthy, J.J. Garcia-Luna-Aceves. A routing protocol for packet radio networks. The First Annual ACM International Conference on Mobile Computing and Networking. ACM, Berkeley, CA, 1995.
[9] C.E. Perkins, T.J. Watson. Highly dynamic destination sequenced distance vector routing (DSDV) for mobile computers. ACM SIGCOMM'94. London, UK, 1994.
[10] Z.J. Hass, R. Pearlman. Zone routing protocol for ad-hoc networks. *draft-ietf-manet-zrp-02.txt*, Internet Draft 1999.
[11] S. Radhakrishnan, N.S.V Rao, G. Racherla, C.N. Sekharan, S.G. Batsell. DST - A routing protocol for ad hoc networks using distributed spanning trees. IEEE Wireless Communications and Networking Conference. New Orleans, USA. 1999.
[12] N. Nikaein, H. Laboid, C. Bonnet. Distributed dynamic routing algorithm (DDR) for mobile ad hoc networks. MobileHOC 2000: First Annual Workshop on Mobile Ad Hoc Networking and Computing. 2000.
[13] OPNET Technologies. OPNET Modeler Product Documentation, Release 11.0 A. 2004.
[14] Ram Ramanathan. Challenges: A Radically New Architecture for Next Generation Mobile Ad Hoc Networks. ACM Mobicom'05. Cologne, Germany. 2005

Yu Wang received his B.S. degree in wireless telecommunication engineering at Harbin Institute of Technology, China. He has years of experience in the telecommunication industry. He is currently a graduate student in Department of Computer Science and Engineering, Florida Atlantic University (FAU). His research interests are wireless ad hoc networks, wireless sensor networks.

Jie Wu is a Professor at Department of Computer Science and Engineering, Florida Atlantic University (FAU). He received the Ph.D. degree in computer engineering from FAU. Dr. Wu has published over 250 papers in various journal and conference proceedings. Dr. Wu serves on the editorial boards of IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS and six other journals. He has served as Program Chair and Program Vice Chair of several international conferences. Dr. Wu was an IEEE Distinguished Visitor.