Cross-layer MAC Enabling Virtual Link for Multi-hop Routing in Wireless Ad Hoc Networks

Kai Hong
Department of ECE
Stevens Institute of Technology
Hoboken, NJ 07030
Email: khong@stevens.edu

Shamik Sengupta
Department of Math. & Comp.Sc.
John Jay College, City University of New York
New York, NY 10019
Email: ssengupta@jjay.cuny.edu

R. Chandramouli
Department of ECE
Stevens Institute of Technology
Hoboken, NJ 07030
Email: mouli@stevens.edu

Abstract—Efficient routing is a fundamental issue in multi-hop wireless ad hoc networks. In this paper, we study the limitation of traditional routing structure in multi-hop wireless ad hoc networks due to (a) the layered structure of a wireless protocol stack and (b) the lack of coordination between medium access control (MAC) and routing protocols. These limitations result in long processing delays in a relay/forwarding node. In order to alleviate these issues, we propose a solution based on cross-layer MAC design, which improves the coordination between MAC and routing layers using an idea we call “virtual link”. The virtual link idea was implemented and tested in an ad hoc wireless network testbed. Experimental results show that the proposed cross-layer design significantly improves the performance in terms of reduced round trip time (RTT), reduced processing time in the intermediate relay/forwarding nodes and increased throughput compared to a legacy architecture.

I. INTRODUCTION

With the advancements of wireless & mobile technologies and the ever increasing demand from the first responders (e.g., fire-fighters, police and emergency medical rescue team) and commercial users for ubiquitous connectivity, wireless ad-hoc and mesh networks are gaining prominence and playing a major role in many applications. For example, during a natural disaster in a region, while it will be inconvenient or even impossible to create an infrastructure-based network, a multi-hop wireless ad-hoc network for first responder communications could be established quickly with the help of low-cost, low-power, multi-functional radio nodes. Unlike infrastructure based networking, multi-hop ad hoc architecture can create wide-area back-haul networks where traffic can flow among the peers directly using relay/forwarding via multiple hops resulting in higher capacity, ubiquitous connectivity and increased coverage.

Efficient routing in multi-hop wireless ad hoc networks is a key fundamental issue to be addressed in order to boost such deployments in first responders or commercial networks. Within multi-hop ad-hoc network, a data packet must traverse through multiple hops from source node to destination node. Additionally, such wireless ad hoc networks differ from their wired counterparts in that their topologies are highly dynamic because of the mobile capability of the nodes. Thus the issue of efficient route discovery/maintenance has been of immense importance. However, even though there has been major focus on routing protocols based on finding efficient shortest route discovery/maintenance or developing link metrics, a key fundamental issue has mostly been unaddressed. Due to the highly complicated nature of Media Access Control (MAC) layer in wireless network, MAC has to be implemented as software. This is different than wired network situation where MAC is implemented as hardware. With such difference, traditional routing structure in wireless network brings in long processing delays for forwarding packet in every intermediate/relay node. As most of the devices in multi-hop wireless ad-hoc network are extremely resource-limited, so traditional routing structure with such unwanted delays will make the system lifetime much shorter and inefficient to operate. Under this situation, without careful design of MAC and routing protocols coordination specifically for multi-hop wireless ad hoc networks, the very features of these networks can turn into disadvantages.

To date, a great deal of literature have focused on routing issues for multi-hop ad-hoc network primarily, along two approaches: (i) shortest route discovery/maintenance and (ii) developing link metrics to analyze a wide variety of performance objectives. At the very beginning, proactive routing protocols like highly Dynamic Destination-Sequenced Distance Vector (DSDV) and reactive routing protocol such as Ad hoc On-Demand Distance Vector Routing (AODV) were designed for solving routing issue and the protocols were developed under simulation environment at initial stage. With the protocols attempted to be built in actual implementation, a large number of issues were encountered such as: the issue of destination never learns of a route to the source node, the way in which RREP packets are forwarded and a rebooted node will lead to routing loops etc. These issues have been discussed in [1] and [2]. With the feasibility issues in the implementation being solved, researchers also focused on performance issue. In [3], a performance comparison between AODV and Dynamic Source Routing Protocol (DSR) are performed. In this work two metrics “Goodput” and “Routing Efficiency” are used for evaluating the performance of these routing protocols. In [4], the energy overhead of DSDV, AODV and DSR are evaluated. This work shows that with different transmission range there is significant differences in energy overhead between these three ad-hoc protocols. In [5], researchers compare energy consumption between different routing protocols including AODV, Flooding Protocol and Low-Energy Adaptive Clustering Hierarchy (LEACH). In
this work, the AODV is considered as not feasible for low energy-limited environment. In [6], a novel routing protocol is proposed to improve energy efficiency. The main idea of this protocol is to reduce message transmissions for routing information update. In [7], cluster-based infrastructure creation are proposed for saving energy. In [8], the research focuses on balancing power consumption between all nodes in a network and concludes that with this effort, the operational lifetime of whole network will be increased. Besides these research efforts in the routing layer itself, there have also been few cross-layer attempts focusing on MAC layer. In [9] [10] [11] [12], the researchers investigate into reduced handshaking/control frames before data transmission and reserved channel resource beyond more than one hop.

However, none of the above-mentioned works focuses on the long processing delays in relay/forwarding node due to the layer structure in wireless communication and the lack of coordination between MAC and routing protocols. In our experiments, for a three hops wireless ad-hoc setup, we observe that 16% of the Round Trip Time(RTT) is spent due to these processing delays; for a four hop ad-hoc network, this is even worse: around 17% of the total RTT. Moreover, it is also observed that at a routing node, an average 40-50% of the processing time is used in transferring data packet between MAC layer and Routing layer. Clearly, these delays result in battery power wastage, higher latency, reduced system efficiency and data rate.

In this paper, we propose a cross layer MAC design enabling Virtual Link for reducing processing time in routing for forwarding nodes. At the same time, the energy consumption for every data frame will also be decreased. Till now, the solution has been successfully implemented based on IEEE 802.11 MAC. In order to measure the performance improvement of this framework in practical environment, we have created a testbed with Soekris NET5501 [13]. With this setup we have evaluated performance improvement from our proposed virtual link enabled MAC to legacy MAC/routing policy [14].

The rest of the paper is organized as follows. We present the proposed cross layer MAC design enabling virtual link in Section II. A detailed discussion on virtual link is also conducted here. In Section III, we present the test bed setup for performance evaluation purpose and discuss the results. Finally, conclusions are drawn in section IV.

II. CROSS LAYER MAC DESIGN ENABLING VIRTUAL LINK

Figure 1 shows the network layer structure and the entire processing path for wired networks like the Ethernet. Here the physical layer and most of the MAC layer functionalities are implemented in network interface card, i.e., in hardware. Such hardware implementation clearly minimizes the impact due to computation-induced delay and power consumption. However, the situation is quite different in a wireless network. Figure 2 shows the same process for a wireless network protocol stack. Due to the complexity of 802.11 protocol, most parts of MAC are implemented as software. [15] shows two type of implementation for 802.11 MAC, FullMAC and SoftMAC. With SoftMAC implementation, most of the MAC layer features are implemented as a module (e.g., in most cases it will be the device driver) in the operating system. Most of the wireless chipset, especially the wireless chipset from Atheros, Intel, Broadcom and Ralink [16], depend on SoftMAC. In FullMAC chipset, like Prism54 [17], the MAC is also not implemented with VLSI but software of chipset, that is firmware.

Routing protocols in the conventional wireless layer structure are typically implemented in network layer. Under this structure, the network layer has three major functions. (i) First, it maintains the routing table, including route discovery, creating corresponding routing entry in the routing table, modifying or deleting routing entries when a route is broken, etc. (ii) Second, it computes the next suitable next hop for the packet. (iii) Third, it re-encapsulates the packets according to the corresponding route entry.

With the traditional routing structure, packets traversing over multiple hops, experience long unwanted processing delays and energy wastage at intermediate routers. Such delays may not be acceptable, especially in critical wireless applications such as first responders’ networks, delay-sensitive real-time media applications etc. Moreover, as most of the devices in multi-hop wireless ad-hoc network are extremely resource-limited, so traditional routing structure with such unwanted delays will make the system lifetime much shorter.
and inefficient to operate. In order to break away from such inefficiencies, we introduce a new “virtual link” concept and propose a new cross layer MAC design which make use of virtual link to forward data packet much more efficiently. Under this proposal the routing layer implements the first function as mentioned above while the second and third functions are implemented in the data link layer.

A. Virtual Link

Before we proceed any further, let us first explain the concept of virtual link in the proposed architecture. Consider an ad-hoc network example as shown in Figure 3. Nodes S and C in this example are within the communication range of each other, i.e., a physical link between nodes S and C exists. Then, these two nodes can communicate with each other over this physical link by using legacy IEEE 802.11 MAC or any other MAC protocol.

In contrast to the physical link between node S and C, the nodes S and D are not within the communication range of each other as shown in the Figure 3. However, note that physical links exist between S-C and C-D. If these two physical links could be combined to obtain a “virtual link” then we obtain a logical link between S and D. It is then possible for nodes S and D to communicate with each other directly minimizing intermediate router processing thereby reducing the delays discussed before. By extending this idea of virtual links, any two nodes in Figure 3 could communicate with each other. The proposed MAC protocol will establish and maintain virtual links automatically and efficiently. With the proposed method, the routing layer will be responsible for the task of maintaining the routing table; however, the task for data frame re-encapsulation will be done by the virtual link enabled MAC. In the following subsection, we will present the system structure of our framework.

B. Proposed System Architecture

In this cross-layer MAC architecture, we introduce two extra modules compared to the legacy MAC. They are Inbound Monitor module (as shown in the workflow of Figure 4) and Self-Learning module (as shown in the workflow of Figure 5). With the newly proposed modules, the steps for creating a virtual link are as follows. When the wireless MAC starts to run in a node, its IP address is noted and the Inbound Monitor also starts to run. The Inbound Monitor of this node checks for the destination IP address on each frame. If the destination IP is equal to its own IP address, this is treated as a normal frame. Otherwise, the Inbound Monitor will look up the corresponding virtual link entry for this frame. If a suitable virtual link is located successfully, this frame will be re-encapsulated according to this virtual link entry and sent to the physical layer immediately for relay/forwarding purpose. If no corresponding virtual link is found, the self-learning module will be triggered. From now on, this monitor module will work on the outbound direction of the IEEE 802.11 MAC. After routing layer re-encapsulate the frame, which triggers the self-learning module, this frame will be shown again on the outbound direction. The self-learning module will create a suitable virtual link according to the new MAC header of this frame. When other data frames arrive at this node, the Inbound Monitor will re-encapsulate the MAC header according to corresponding Virtual Link entry. Figure 4 shows the workflow of Inbound Monitor module. Figure 5 describes the workflow for the Self-Learning module.

In this regard, we introduce two new parameters: an integer value \( T_e \) and a boolean value \( MRU \). \( T_e \) is used to decide the expiry time for a virtual link entry. If a virtual link entry is not activated by a data frame for \( T_e \), this entry will be removed from Virtual Link Table. The Boolean value \( MRU \) is used for maintaining the Virtual Link table. With \( MRU \) set to TRUE, MAC will monitor the Routing Protocol Data Unit (RPDU) on both inbound and outbound directions; when
Is this node working in self study process?

Yes

Process like legacy MAC

Fig. 5. Work flow of Self-Learning module

Is this the frame which triggers self-study process?

Yes

Create Virtual Link for this Destination

No

Yes

III. TEST BED SETUP AND EXPERIMENT RESULTS

For the purpose of evaluating the performance of our proposed MAC structure, we conducted extensive experiments and present the results in this section. We primarily focus on three metrics - processing time in MAC layer and Routing layer, round trip time (RTT) for data packets with different length and hop counts and throughput which is measured at destination node.

A. Test bed Setup

In order to setup a test bed, we are using Soekris NET5501 [13] as our network nodes. We setup our multi-hop ad hoc networks in a region of 20 m × 20 m area with wireless nodes (Soekris NET5501) scattered randomly in the region at a distance of 5–20 meters from each other. NET5501 is equipped with a 433MHz AMD Geode LX (X86 architecture) CPU, 512MB DDR-SDRAM. Compact Flash card is used for software and data storage purpose. A customized Linux 2.6.24.6 is deployed in our nodes. IPTABLE is applied for blocking the communication for simulating designed topology. The wireless transceiver for this node is an Atheros AR5414 chipset based wireless network interface card (WNIC). We have implemented our proposed MAC with the driver (Madwifi) of this WNIC. ICMP and IPERF are used for measuring the performance of our proposed MAC structure.

B. Experiment Result

In order to find out the processing time consumed in coordinated MAC/Routing Layer, we create monitors on both inbound and outbound direction of MAC. With the time difference in these two monitors, the time consumption for MAC and Routing layer are obtained. Figure 6 shows the result, X-axis for different packet size and Y-axis for time consumption. As observed from the Figure 6, with our proposed approach, we get a 45% improvement with 100 bytes long packet over the legacy routing mechanism while with the packet size going up to 20,000 bytes, we get an overwhelming 55% improvement. This clearly demonstrates that cross-layer MAC design enabling virtual link increases the efficiency of multi-hop routing by sufficiently reducing the processing time consumption at the intermediate relay nodes.

Next, we focus on the round trip time (RTT) performance comparison for data packets with different length and hop counts. In these experiments, ICMP packets with different size are used for measuring the Round Trip Time (RTT). Figure 7, 8 and 9 show RTT comparison between AODV-UU with legacy MAC and AODV-UU combined with the proposed MAC. As shown in the results, proposed methodology clearly reduces the round trip time over the legacy approach. Moreover, it is also interesting to note that with the increasing hop counts the performance improvements are also increasing thus clearly proving the proposed methodology as ideal for multi-hop wireless ad hoc networks.

Figure 10 shows throughput difference between AODV-UU with legacy MAC and AODV-UU with our proposed MAC. In this experiment, an iperf server runs in source node. The throughput is measured from the iperf client side which runs in
destination node. In this figure, x-axis presents hop counts, and y-axis shows the throughput, which is measured at destination node. As shown in the Figure 10, the throughput improvement of the proposed approach over the legacy approach is around 7% with 2-hops case and around 10% with 4 hops setup thus clearly justifying the basis of our proposed methodology.

IV. Conclusion

A cross-layer MAC design based on the concept of virtual link is proposed in this paper for improving the routing efficiency in multi-hop wireless ad hoc networks. We introduce two new modules: Inbound Monitor and Self-Learning that help in enabling and maintaining the virtual links. To demonstrate the proposed method, we have implemented a wireless ad hoc testbed system and compared its performance with a traditional legacy routing technique. The tests performed in this testbed clearly demonstrate that cross-layer MAC design employing the proposed virtual link concept reduces the processing time at the intermediate nodes by approximate 50% while the throughput increases by 7–10% when compared with the legacy routing algorithm.

ACKNOWLEDGMENT

This research was partially funded by NSF CNS #0917008, NSF CCF #0916180 and NIJ #2009-92667-NJ-IJ.

REFERENCES