A Unified MAC and Routing Framework for Multi-Channel Multi-Interface Ad Hoc Networks

Jinhua Zhu, Xin Wang, *Member, IEEE,* and Dahai Xu, *Member, IEEE*

***Index Terms*— Ad hoc networks, multi-channel, multi-radio, MAC, routing, cross-layer**

***Abstract*— Improving capacity of wireless networks is critical and challenging. Although wireless standards such as IEEE 802.11 allow for the use of multiple channels at physical layer, current MAC and routing protocols of mobile ad hoc networks are mainly developed for running over one channel. In this paper, we design a unified MAC and routing framework to exploit the temporal and frequency resources to significantly improve the throughput of ad hoc networks. Our joint channel assignment and routing scheme searches for an efficient transmission path taking into account the constraints due to the limited number of available channels and radio interfaces and the impact of MAC layer scheduling. Channel maintenance schemes are proposed to adapt the path and channel assignment in response to the changes of network topology and channel condition, and feedbacks from the MAC layer. Given the routing path and channel assignment, our scheduling scheme at the MAC layer explores the resources at time domain to coordinate transmissions within an interference range to maximize channel usage, reduce channel access competition among nodes assigned the same channel, coordinate radio interface usage to avoid unnecessary channel switching, and support load balancing. Complemented with the scheduling algorithm, a prioritized transmission scheme is presented to resolve collisions from multiple nodes scheduled to transmit on the same channel in the same time period, and reduce transmission delay of mission critical packets and message broadcast, which help to further improve network performance. Our simulations demonstrate that our integrated MAC and routing design can efficiently utilize the channel resources to significantly improve the throughput of multi-channel multi-interface ad hoc networks.**

I. Introduction

Mobile Ad Hoc networks (MANETs) are important in vehicular communications and communications in military and disaster rescue environments. With the popularity of wireless devices and the ever-increasing throughput demand of applications, it is critical to develop protocols that can extract the highest level of performance using the available spectrum. Although wireless LAN standards such as IEEE 802.11 often allow for transmissions on multiple physical channels, current MAC and routing protocols in infrastructure-free ad hoc networks are generally designed to transmit data only on one channel. Also, most existing wireless devices are equipped with only one wireless interface, with which a node can transmit or listen to only one channel at a time. On the other hand, although a node equipped with multiple radios can potentially communicate with several neighbors concurrently using different channels to improve the throughput, the need to reduce equipment size and cost restricts the maximum number of radios a node can have. It is more efficient for wireless devices to transmit on all the available channels with a limited number of radio interfaces. The objective of this work is to develop a unified MAC and routing framework for mobile ad hoc networks to fully exploit the benefits enabled by multiple channels with a small number of radio interfaces.

There are many challenges in designing an efficient scheme for interface management and channel allocation in a practical multi-channel multi-interface (MCMI) environment. As the number of orthogonal channels is limited, more than one node in a neighborhood could contend to access the same channel. Careful channel assignment is needed to control the load at a channel and reduce the collisions. When the number of interfaces is smaller than the number of channels, it requires a careful channel usage coordination for two nodes to tune to the same channel for communication without incurring a large interface switching delay. In addition, there is a need to increase concurrent transmissions in a neighborhood over different radio channels. Besides these issues, in a multi-hop network, it is critical and challenging to establish a routing path that exploits MCMI feature for a better throughput, and maintain the path to cope with the increased interference and route inefficiency due to the environmental change and node movement. It is also important to support efficient broadcast in a multi-channel environment.

Since the above issues span the physical, link and network layers, a *cross-layer approach* is called for. Accordingly, we will develop *a unified MAC and routing framework* to accomplish our main objective: exploiting multi-channel multi-interface capabilities in mobile ad hoc networks to fully use the available spectrum to improve the network performance. Our framework jointly considers routing, channel assignment, as well as scheduling and prioritized transmission. At the routing layer, our new *link cost model* captures the characteristics of MCMI networks and the impact of MAC layer scheduling, and a *joint channel assignment and routing scheme* concurrently searches for the minimum cost path and assigns channels to nodes on the path. Our route maintenance scheme adapts the path and channel assignment based on changes of topology and channel condition, and feedbacks from MAC layer. Given the channel assignments during path setup, a *scheduling scheme* is used at MAC layer to coordinate the channel usage and interface sharing/switching to enable communications between nodes, as well as to reduce channel access competition, transmission confliction and unnecessary interface switching. Finally, the transmission priority is used to enable timely transmission of control packets through broadcast and delay sensitive packets.

Without loss of generality, we assume that the number of

Copyright (c) 2010 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

Jinhua Zhu is with Ericsson Inc, Boulder, CO 80301; Xin Wang is with the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794, USA; Dahai Xu is with AT&T Research Lab, Florham Park, NJ 07932, USA.

The project is partially sponsored by US NSF under grant numbers CNS 0751121 and CNS 0628093.

interfaces is smaller than the number of available channels. Our contributions can be summarized as follows:

- Designing an efficient routing metric that can track the rate diversity at different links, the transmission failures due to collisions, the constraints due to interface sharing, and the channel competition due to the limited number of channels.
- Developing a joint route discovery and channel assignment scheme to exploit the capability of multiple channels and multiple interfaces to minimize the interference among neighboring nodes, and thus maximize the number of possible concurrent transmissions.
- Incorporating a channel and route maintenance scheme to adapt the routing path and channel assignment to catch the topology and interference changes due to node movement and to balance channel and interface usage.
- Designing a scheduling scheme which manages resources in time dimension to coordinate channel usage and interface sharing among neighboring nodes assigned the same channel to reduce channel competitions, avoid transmission confliction due to uncoordinated transmissions from multiple nodes to the same receiver at the same time, and to minimize the effect of channel switching delay due to the uncoordinated random access of different channels. Our scheduling scheme can also support load balancing and enable fairness among neighboring nodes.
- Enhancing the 802.11 MAC protocol with prioritized transmitting to further resolve collisions among nodes scheduled to transmit on the same channel in the same time period, reduce multi-channel broadcast delay and the transmission delay for mission critical applications, and allow unscheduled nodes to opportunistically use the available channel resources to improve throughput.

Multi-channel multi-radio wireless networks have received a substantial amount of recent interest, especially in the context of wireless mesh networks. The schemes proposed for static wireless mesh networks [1]- [6] often require offline solutions and are generally difficult to be used in or not applicable to mobile ad hoc networks. Although a large number of efforts have been made to design MAC schemes to coordinate channel usage in ad hoc networks [7]–[12], there are very limited routing designs [13]–[15]. As the interference range is generally much larger than the transmission range and there is a coupling between transmissions in different neighborhoods in a large network, simply considering local-range channel assignments and transmissions is inefficient. On the other hand, decoupling routing and channel assignment [14] cannot capture the interference along the transmission path, while using single interface [13] in multi-channel environment for routing would result in poor connectivity.

To the best of our knowledge, our work provides the first practical network framework that concurrently considers routing and channel assignment at network layer as well as scheduling and prioritized transmission at the MAC layer to support efficient communications over MCMI ad hoc networks. Different from literature studies, our algorithms are completely distributed without assuming the knowledge of network parameters and traffic load in advance, and consider the practical limitation in the number of channels and interfaces. Instead of assigning channels to the links, our scheme assigns receiving channels to nodes to allow more freely and concurrent transmissions in different channels and avoid the deafness problem when a transmission pair tunes their radio interfaces to the same channel at different time. The channel assignment is performed during path setup to better coordinate channel usage in a larger network range for a longer time, and adapts during path maintenance to reduce interference. In addition, our scheduling scheme coordinates transmissions in time domain to constrain the number of concurrent transmissions in a channel and coordinates radio interface switching to avoid transmission conflict. Moreover, our prioritized transmission scheme reduces delay of mission critical traffic and control messages.

The rest of the paper is organized as follows. We discuss the literature work in Section II, and provide a system overview in Section III. In Section IV, we present the problems pertaining to a multi-channel multi-interface network, and describe our scheduling algorithm and the prioritized transmitting scheme to address these issues. In Section V, we introduce a new routing metric, based on which we describe in details a joint routing and channel assignment scheme, and an efficient channel and route maintenance scheme. Section VI describes our evaluation using simulations. We conclude the paper in Section VII.

II. Related Work

Several efforts [7]–[12] have been made to modify the MAC protocols to support multiple channels. Wu *et al.* [9] employ two transceivers, while the dedication of one channel for control messages would result in poor channel utilization when the number of channels is small or control channel bottleneck when the number of channels is large. The schemes in [7], [8] require the number of transceivers at each node to be the same as the number of channels, which are thus very expensive. In [10], [11] the authors propose multiple access schemes for the nodes equipped with single interface. RICH-DP [12] is a receiver-driven scheme that requires all nodes to use a common frequency hopping sequence. A centralized algorithm is proposed in [16] to consider congestion and channel allocation, while the scheme in [17] targets to address the starvation problem in a CSMA-based multi-hop wireless network.

Predominant routing protocols such as DSR [18] and AODV [19] are purely based on the shortest path metric without exploiting the capabilities of multiple channels [20]. The routing protocol in [13] considers single interface for multiple channels, which results in poor connectivity since a node can only transmit or receive in one channel at a time. In [14], the channel assignment is done prior to routing, which ignores the fact that channel assignment and routing are inherently inter-dependent and transmission on the same path may experience intra-channel interference.

Recently, several schemes have been proposed to utilize multiple channels in static wireless mesh networks [1]- [6] where all the traffic is directed toward specific gateway nodes. These schemes are difficult to be applied in the mobile ad hoc networks which require a distributed scheme to react quickly to topology change. The scheme proposed in [21] combines multi-channel link layer with multi-path routing. Although interesting, many design ideas (e.g, super-frame pattern, dynamic adjustment of T/R ratio and multi-path routing) proposed in the paper target to address the inefficiency due to the half-duplex transmissions as a result of using one radio interface at each node. Use of a single interface would lead to more severe multi-channel hidden terminal problem [10] and deafness problem. In [20], the authors extend the work from [22] and propose a new routing metric, WCETT, to select channel diversified routes in wireless mesh networks with the assumption that the number of interfaces per node is equal to the number of channels used in the network. The proposed routing metric only considers intra-path interference. Instead, our scheme is designed to handle the more general case that the number of interfaces may be smaller than the number of available channels. Assuming the channel has been assigned, the work in [23] considers queueing delay in the routing metric. Although it may be good to consider load, the dynamics of queue status may lead to routing instability. Instead, we consider load balancing at MAC layer during scheduling, which can better handle traffic dynamics.

The authors in [15] perform theoretical studies on channel assignment, scheduling and routing without considering a practical protocol design to implement the algorithms. Although the proposed scheme is not centralized, a super node is implicitly assumed to perform the optimal channel assignment and scheduling in each neighborhood. It may involve a high control overhead to distribute necessary information and perform channel assignment in each time slot, and it is not clear how nodes in different neighborhood could coordinate in channel usage. An even higher overhead would be incurred to collect end-to-end queue information in each time slot to perform routing in alternative paths. In contrast, we propose a comprehensive routing metric to capture the limitation in the number of available channels and radio interfaces as well as interference and transmission conflict for efficient path setup and channel assignment in a MCMI network. The scheduling algorithm is purely distributed and each node can make scheduling decision to efficiently coordinate channel usage and interface switching without need of complicated signaling messages.

III. System Overview

The goal of our work is to design an efficient MCMI communication framework with integrated MAC and routing for mobile ad hoc networks. The proposed schemes exploit resources both from *frequency* domain through channel assignment and *time* domain through transmission time slot scheduling to significantly increase the network throughput. Our design at the routing layer includes: 1) a *link cost model* to capture the characteristics of MCMI networks and the impact of MAC layer scheduling, 2) a *joint channel assignment and routing scheme* to concurrently search for the minimum cost path and assign channels to nodes along the path, 3) a route maintenance scheme to adapt the path and channel assignment in response to changes of network topology and channel conditions and MAC feedbacks. Given channels assigned during the path setup, our design at the MAC layer includes: 1) a *distributed scheduling scheme* to coordinate the channel usage in the unit of time slot to reduce competition among nodes assigned the same channel within an interference range, and coordinate interface sharing and switching to reduce transmission conflict and unnecessary switching delay; 2) a *prioritized transmission scheme* to coordinate multiple nodes in accessing a specific channel given the scheduled channel usage within a time slot to improve network throughput while reducing the delay of high priority control and data packets.

In a multi-channel network, a communication may fail if an intended receiver is currently tuned to a different channel, resulting in a deafness problem. To avoid this problem, in the proposed MCMI system, we ascribe the radio interfaces to two types, *listening interface* (i.e., *LI*) and *transmitting interface* (i.e., *TI*). During path setup, one radio interface of a node will be designated as *LI* and assigned a channel, called *listening interface's channel* or *LIC*. A node uses its LI to constantly monitor the conditions of the assigned LIC and intercept the packets targeted to the node, which avoids the deafness problem. The other interfaces of a node are called transmitting interfaces which can be flexibly tuned to different channels assigned to its neighbors to transmit data packets.

In our design, two types of messages are used for updating channel status. A *Hello Message* will be sent by a node periodically to maintain network topology, as generally done in other routing protocols. To reduce the interference among the competing nodes on a channel, it is helpful to have information on network topology and channel assignment of nodes within an interference range. The interference range can be multiple times the transmission range, and the interference reduces quickly as the distance between the transmitter and receiver increases. To reduce the implementation overhead, in this work, we consider the interference up to two hops [20], thus, a *Hello Message* carries its one-hop neighbors' information. In addition, a *Channel Update Message* will be sent within the interference range when the channel assignment for a node is changed.

In explaining our design, each node is assumed to have two interfaces. However, our design can be extended to support more radio interfaces, with one interface designated as *LI* and the other interfaces serving as *TIs*.

IV. MAC protocol

In our MAC design, a channel and interface scheduling scheme coordinates node transmissions in a neighborhood, which is complemented with a prioritized channel access scheme to improve transmission efficiency while reducing delay of important control and data packets. Our MAC scheme addresses the following issues:

1) *Interference among the transmissions over the same channel.* There is generally a limited number of channels in the system. Due to cost, time and policy constraints, the number of channels a node can tune to and monitor is limited. Therefore, multiple nodes in a neighborhood may have to use the same channel, incurring competitions in channel access and interference among concurrent transmissions.
2) *Interface switching delay*. A node generally has a fewer number of radio interfaces than the number of available channels. To explore use of multiple channels, an interface needs to be switched among different channels. As channel switching incurs a non-ignorable delay [11], it would be more efficient to reduce channel switching.
3) *Transmission conflict*. A node may have several downstream nodes listening to different channels. Without any coordination, independent transmissions from multiple upstream nodes to the same channel will result in collisions, while a better channel usage coordination would lead to concurrent transmissions. For example, in Fig. 1, node A can transmit to node C and D using Channel 1

and 2 respectively, while node B can transmit to node D and E using Channel 2 and 3 respectively. Without any coordination, node A and B may try to transmit to node D using Channel 2 at the same time while neither Channel 1 or 3 is used, which causes both collision at the same receiver and channel resource wastage.

4) *Broadcast delay*. As different nodes may be listening to different channels, to reach all potential neighboring nodes, a broadcast packet needs to be transmitted in each channel one by one. There is also a delay in switching interface between channels and a random access delay for a node to win the competition in channel access. This would add up to an extremely high broadcast delay, which results in a high path setup delay (to broadcast route searching messages), throughput degradation and even routing failure (due to delayed channel state updates).

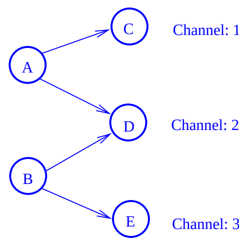

Fig. 1. Example of Transmission Coordination

A. Channel Scheduling Scheme

In a MCMI system, a simple exchange of RTS/CTS between a sender and a receiver on the listening channel of the receiver is not enough to avoid the hidden terminal problem as a potential interference node may be listening to a different channel, while sending a RTS/CTS to all channels of neighbors before each packet transmission would incur a high overhead. Instead, we design a slot-based distributed scheduling scheme to reduce the number of interface switching at each node, coordinate transmission to reduce the node contention in accessing the same channel, and resolve transmission confliction. We define a *time slot* to be the duration a node is scheduled to use a channel for receiving. Our scheduling has the following procedures: 1) When multiple nodes within the interference range are assigned the same listening channel, only one node is scheduled to receive in a time slot; 2) When a scheduled receiver has multiple upstream nodes, only one of the nodes will be scheduled to transmit; 3) When a node is scheduled to transmit to multiple receivers with different listening channels, it will select one of the receivers to transmit packets. Instead of selecting only one node to access a channel, as analyzed in Section V-A.2, our scheduling algorithm only *constrains* the number of nodes that can transmit on a specific channel in a time slot. This design avoids the need of strong synchronization among nodes and takes advantage of multiplexed transmissions from multiple nodes to improve throughput. For multiple nodes scheduled to transmit on the same channel in a time slot, a priority-based collision avoidance scheme (Section IV-B) is used to further coordinate the transmissions. By constraining the number of nodes in channel competition, however, our scheduling scheme can avoid a significant throughput degradation under heavy load as in a pure CSMA/CA-based scheme such as 802.11.

For efficient scheduling, it is important to select an appropriate slot length to reduce the impact of switching delay while not introducing a significant waiting delay for other nodes not scheduled for transmission in a slot. In the proposed MAC scheme, only slot-level synchronization is needed among neighboring nodes, and a global synchronization is not required. As RTS/CTS will be used for handshaking before each packet transmission in our collision avoidance scheme, strict synchronization is not necessary. We consider the interference range up to two hops [3], and the nodes to transmit on the same channel within the interference range as *contending entities.* With periodic transmission of *Hello Messages* and triggered sending of *Channel Update Messages* within a two-hop neighborhood, every entity knows the set of its contenders. For an entity i, a contention resolution algorithm must decide whether i is the winner in a *contention context*, and every other contender must yield to i whenever i derives itself as the winner. As the data packet from the sender to the receiver is generally longer than the confirmation packet from the receiver to the sender, it is more important to reduce interference at the receiver side. Our scheduling has two phases, *receiver scheduling* and *transmitter scheduling*. During receiver scheduling, we consider the receiving nodes within an interference range as the contending entities, and our algorithm will schedule at most one node to receive packets on a given channel within the interference range. During transmitter scheduling, all upstream nodes of a scheduled receiver are considered as contending entities, and one node will be scheduled for transmission in a time slot.

It is critical to reduce control overhead during scheduling. In our receiver scheduling, a node *self-determines* if it is scheduled for receiving in a slot based on the knowledge of local network topology and channel assignment *without need of signaling messages*. To derive a unique winner in a time slot t, a candidate receiving node generates a priority number for itself and each of its contending nodes, i.e., the nodes assigned the same receiving channel within the interference range. If the node's priority number is the highest, it is scheduled for receiving. For simplicity, the priority of a contending entity X can be set to a random number $Rand(X, t)$ with a value between 0 and 1. If more than one contending entity have the highest priority, the one with the largest ID will be selected.

This algorithm is summarized as Algorithm 1, with i denoting the node ID of the potential receiver, t denoting the time slot, and $N_{ch,i}^{2-hop}$ denoting node i's two-hop neighbors contending for the same *LIC* (ch) as i. $Rand(X, t)$ is adopted from the $Hash()$ function used in [24]:

$$Rand(X, Y) = Hash(X \oplus Y)/2^{64} \tag{1}$$

where Hash(x) is a fast random integer generator that hashes the input argument x to an integer, and $\oplus$ is the concatenation operation on two operands. We assume the size of the output of $Hash()$ function is 64 bits. Node i will win the competition and be scheduled for receiving in slot t if it has the highest priority, otherwise, it yields to other competing nodes.

A scheduled receiving node may have several senders. To avoid transmission confliction, each candidate sender *self-determines* if it is scheduled to transmit in a time slot without signaling. The algorithm works as follows. When a node R is assigned a new receiving channel, it broadcasts a *Channel Update Message* to notify all the potential senders the identifiers

Algorithm 1 $ReceiverScheduling(i, ch, t)$

```
for (all j ∈ N_{ch,i}^{2-hop} ) do
    if Rand(i,t) < Rand(j,t) then
        return FALSE
    end if
end for
return TRUE
```

of its two-hop neighbors which share the same *LIC* with R. Knowing the two-hop neighbors of all its targeted receivers, at the beginning of each time slot, a node S checks if any of its receivers are scheduled using Algorithm 1. If it finds one or more nodes are scheduled for receiving, node S will check whether it is scheduled to transmit packets to the scheduled receiver(s), using Algorithm 2. To avoid transmission contention and balance the load among sending nodes, a receiver i will assign non-overlapping *probability range* $P_{i,j}$ for each of its upstream node j based on j's current traffic load to i. A sending node generates a random value based on the receiver's ID and the time slot number. If the random value falls into the range assigned to the node, the node has the highest priority for transmission among all the competing senders. In case a node is scheduled for transmitting to more than one receiver, it can randomly pick one to transmit during the scheduled slot.

Algorithm 2 $SenderScheduling(i, ch, t)$

```
if (Rand(i,t) ∈ P_{i,j}) then
    return TRUE
else
    return FALSE
end if
```

An example is shown in Fig. 2 to explain how our scheduling works. There are four senders (Node A, B, C, and D) and three receivers (node E, F and G). Assume that all the receivers are within interference range and are assigned the same receiving channel. At the beginning of a time slot q, each sender will check whether it is scheduled for transmission based on its probability range and the receivers' priority calculated according to Eq. 1, which are shown in Fig. 2. For example, node A first checks whether node E is scheduled for receiving during slot q by comparing the priority values of all the receivers within node E's interference range. Since node E's priority value (0.4) is the highest among all three receivers, node A can decide that node E is scheduled for receiving. Node A then checks whether it is scheduled for transmitting to node E. As node E's random value (0.4) falls in node A's probability range ([0 : 0.5)), node A determines that it is scheduled to transmit to node E during slot q. Similarly, node B determines that node E is scheduled for receiving but it itself is not scheduled to transmit to node E. Node C and D determine that node F and G are not scheduled for receiving during slot q.

To balance the load of the potential senders, a simple formula would be used to assign the probability range proportional to the average queue length of the senders. A sender can report its average queue length to the receiver through RTS or by piggybacking with the data packets. The average queue length $\hat{L}_k(t)$ of a sender k can be calculated with Eq. (2):

$$\hat{L}_k(t) = (1 - \alpha) \cdot \hat{L}_k(t-1) + \alpha \cdot L_k(t), \qquad (2)$$

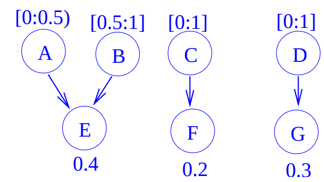

Fig. 2. Example of Scheduling

where $L_k(t)$ is the current queue length and α is a *memory factor*. Assume a receiver r has M senders, the probability range for a sender k can be calculated as:

$$P_{r,k} = \begin{cases} [0, \frac{\hat{L}_1}{L}) & \text{if } k = 1 \\ [\frac{\sum_{i=1}^{k-1} \hat{L}_i}{L}, \frac{\sum_{i=1}^{k} \hat{L}_i}{L}) & \text{if } 1 < k < M \\ [\frac{\sum_{i=1}^{M-1} \hat{L}_i}{L}, 1] & \text{if k = M} \end{cases}$$

where $L = \sum_{i=1}^{M} \hat{L}_i$. When the queue length of a sender is unknown, i.e., when a path is first set-up, the sender will be assigned default transmission range $[0, 1/M)$, and the remaining $M-1$ senders will be assigned range proportional to their queue length within $[1/M, 1]$. To reduce instability, the adjustment of probability should not happen frequently as a large queue length may be caused by some traffic bursts and the adjustment itself involves additional overhead. The transmitter scheduling scheme attempts to give the node with a higher load the higher priority for transmission. There is no need to have accurate queue lengths to calculate the probability range. In case more than one node are scheduled to transmit to the same receiver due to inaccurate range information at nodes, the scheduled nodes can compete in channel access using our priority-based collision avoidance scheme discussed next.

B. Prioritized Transmission

The proposed scheduling scheme coordinates channel switching, resolves transmission confliction from several senders to the same receiver and constrains the number of nodes within an interference range that would contend for the same channel during a time slot (see Section V-A.2). With the support of time-slot based scheduling, there are still additional issues to address: 1) There is a need to coordinate transmissions from multiple scheduled nodes on the same channel; 2) The nodes scheduled for communications may not have enough data packets to fully utilize the time slot assigned, and to improve the throughput, it is desirable to allow other nodes to use the spare time slot; 3) Mission critical data packets have tight delay requirements; 4) It is desirable to reduce broadcast delay to deliver important control information in time. To address all these issues, we complement the scheduling scheme with a prioritized transmission scheme with three levels of priority:

The *first* (highest) level of priority is given to some important packets that need to be transmitted as soon as possible, such as some routing control packets (e.g., RREQ, RRER and RREP packets) and mission critical data packets. To avoid collision in transmitting the first priority packets, each node waits for some random time within a window $W0$.

The *second* level of priority is given to the packets from the scheduled senders to the scheduled receivers. The sender also waits for some random delay before transmitting a RTS packet but with a different delay window $W1$ larger than $W0$.

The *third* level of priority will be assigned to the non-scheduled senders to avoid wasting the time slots that cannot be used up by the scheduled transmissions. To avoid competing with the scheduled sender, a non-scheduled sender can wait for the entire window $W1$ and an interval equal to a RTS/CTS transmission, and then transmit after a random delay within some window $W2$. After the first successful transmission, the non-scheduled nodes only need to wait for a random period of time within the window $W2$ before transmitting subsequent packets. In addition, a non-scheduled sender should reset the timer and wait for $W1$ period first once detecting a transmission from a scheduled sender, so that the scheduled sender still has higher priority in the remaining time slot.

It is worth mentioning that our scheme is robust in presence of scheduling error due to incorrect or outdated topology information. If a sender mistakenly determines that it is scheduled for transmission in one time slot, it will compete with other scheduled senders by using RTS/CTS scheme. On the other hand, if a sender wrongly decides to yield to other nodes, this time slot will be used by other scheduled or non-scheduled nodes with a lower priority. We will show in the next section that more than one node within two-hop neighborhood can be scheduled for transmission within a time slot.

V. Channel Assignment and Routing

Existing routing protocols for wireless ad hoc networks [18] [19] generally use hop count as the link cost without considering the effect of multiple channels on path establishment and transmission performance. For example, there are two possible paths ($SABD$ and SCD) between node C and D in Fig. 3. Assume each link has the same transmission rate. Although path SCD has only two hops, as node C and D are assigned the same listening channel ($ch2$), the two links SC and CD cannot be used to transmit packets at the same time. Therefore, packets from node S may transmit faster along path $SABD$ to node D. However, this comparison is based on a random channel assignment. If the channels for node C and D can be reassigned to different ones during path setup to avoid interference on two contiguous links, then the path SCD would lead to lower delay. In this work, we design a channel assignment and routing protocol to explore the benefits of multiple channels and multiple interfaces, while mitigating the constraints due to the limited number of radio interfaces and channels.

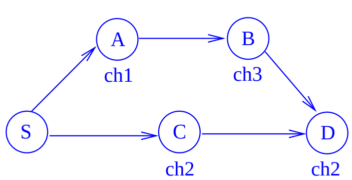

Fig. 3. Two possible paths

A routing protocol generally searches for the minimum cost path between the source and the destination. As the cost of a link is affected not only by the channel assignment for the link itself but also by the channel assignments for other links within an interference range, finding the minimum cost path usually involves a non-linear optimization process, which would make it difficult and unrealistic to find the theoretical optimal path in mobile ad hoc networks. Instead, our routing protocol adopts a "greedy" algorithm to quickly find a sub-optimal path. This routing scheme can also be easily implemented.

In this section, we first introduce our new link cost model, and then describe how an efficient routing path can be established using the new cost model.

A. Link Cost Model

Link cost plays an important role in the routing protocol. We choose delay as the link cost because it is closely related to the throughput. A short end-to-end delay will generally improve throughput. We consider some important factors that impact the link delay as follows.

1) Interface Capacity: In wireless networks, different interfaces may have different capacities (e.g. 11Mbps in IEEE 802.11b and 54Mbps in IEEE 802.11a/g), which result in different transmission delays for the same packet. Therefore, we can define a *Transmission Delay Factor* (f_t) as $f_t = 1/W$, where W is the link rate, and a higher rate would lead to a lower delay over the link.

2) Retransmission and MAC Scheduling: Retransmission due to packet loss and error will increase the overall transmission delay. The packet error rate of a link in a channel can be measured [20]. However, as a node generally has fewer interfaces than the available number of channels, it is difficult to measure the packet error rate in real time for every channel. In order to measure the condition of a channel, there is also a need to transmit data on the channel first, which may not be possible before the channel is assigned. The interference measurement in [25] can be only used for static networks. Instead, we estimate the packet error rate analytically based on our scheduling scheme.

Assume the interference range is about twice the transmission range. In our scheduling scheme, only one receiver is scheduled within a two-hop neighborhood. Assuming the network area is A, the transmission range is R and the nodes are evenly distributed. If the scheduled receivers are in the center of the adjacent circles with a radius R, the maximum number of scheduled receivers on a specific channel in the whole network is $N_r = A/\pi R^2$. For each scheduled receiver, there is only one corresponding scheduled sender. Thus the maximum number of scheduled senders in the network on a channel is: $N_s = N_r$. Assuming that all senders are also evenly distributed, the average number of *contending senders* in the two-hop neighborhood of a receiver can be calculated as

$$N_s^{2-hop} = (N_s/A) \cdot (\pi(2R)^2) = 4, \tag{3}$$

which is independent of the node density in the network. The contending nodes will compete in channel access, and resolve collision through RTS/CTS similar to 802.11 as described in Section IV-B. Most transmission failures are due to collisions (e. g., collisions in RTS messages). For a 802.11 network, the collision probability or packet error rate (p) is impacted by the number of contending nodes (n) [26]:

$$p = 1 - \left(1 - \frac{2(1-2p)}{(1-2p)(\widetilde{W}+1) + p\widetilde{W}(1-(2p)^m)}\right)^{n-1}, \tag{4}$$

where $\widetilde{W} = CW_{min}$ and $m = \log_2(CW_{max}/CW_{min})$.

As our scheduling algorithm restricts the average number of competing nodes within the interference range to be a constant number 4, based on Eq. 4, the average packet error rate (p) is small and a constant. The expected number of transmissions

(*ETX*) can be calculated as $\frac{1}{1-p}$. The larger the expected number of (re)transmissions, the higher the delay in one link. Therefore, *ETX* can be used as the retransmission delay factor (f_r)

$$f_r = \frac{1}{1-p}. \quad (5)$$

Since p is a constant, f_r also has a constant value. Although the channel condition is not considered during channel assignment time, the channel condition will be considered when there are active transmissions on the channel, and the channel can be changed through the maintenance strategies discussed in Section V-C if significant errors are detected.

3) Limited Number of Channels: When there is a limited number of channels, nodes in a neighborhood may be assigned to the same channel. While scheduling helps to mitigate contention on the same channel, it also introduces delays. Generally, node A can communicate with node B only if node B is scheduled for receiving and node A is scheduled for transmitting to node B. In our scheduling scheme, among the nodes sharing the same listening channel (*LIC*) within a two-hop neighborhood, only one node is scheduled for receiving in a slot. Assume each node has the same probability of being scheduled for receiving and node B is assigned channel ch as its *LIC*, then the probability of node B being scheduled for receiving in channel ch is

$$p_r(B) = \frac{1}{N_{B,ch}^{2-hop}}, \quad (6)$$

where $N_{B,ch}^{2-hop}$ is the number of nodes sharing the same listening channel ch and within B's two-hop neighborhood.

Assume each upstream node (potential sender) has the same probability of being scheduled for transmitting to a scheduled receiver, and N_{ToB} is the number of upstream nodes of node B, then the probability of node A being scheduled for transmitting to node B can be defined as

$$p_t(A \to B) = \frac{1}{N_{ToB}}. \quad (7)$$

Therefore, the delay factor (f_s) between node A and B due to scheduling of transmission as a result of a limited number of channels is

$$f_s = \frac{1}{p_r(B)} \cdot \frac{1}{p_t(A \to B)} = N_{B,ch}^{2-hop} \cdot N_{ToB}. \quad (8)$$

This factor reflects the impact of network topology and channel constraint on network throughput. If there are a large number of nodes sharing the same *LIC* as the receiver within the interference range and/or when the receiver has many upstream nodes, there will be a higher transmission delay through the corresponding link. The routing protocol should avoid such receiver nodes during path searching.

4) Limited Number of Radio Interfaces and Scheduling Conflict: To reduce node size and implementation cost, a node generally has fewer number of radio interfaces than the number of radio channels of the network, which may lead to extra delay for interface usage coordination. If node A has several downstream nodes, as scheduling is performed distributedly in reference to each receiver, it may be scheduled for transmitting to more than one receiver in a time slot. For example, in Fig. 4, node A has three downstream nodes B, C and D which are scheduled to receive on channel 2, 3, and 4 respectively. Node A is also scheduled to transmit to all the three nodes. Since it can only transmit to one node at a time, some scheduled time slots are wasted leading to a higher average link delay. To evaluate the cost due to the conflicted scheduling, we calculate p_{AB}, the "equivalent" fraction of the time slot scheduled for node A to transmit to node B that node A can eventually use to transmit packets to node B. The lower the "equivalent" time fraction, the higher the delay.

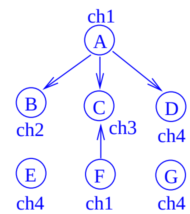

Fig. 4. Transmission Conflicting Example

The concept of "equivalent" fraction of the time slot can be understood in an intuitive way. We assume the scheduling in each channel is independent and node A will randomly pick one channel to transmit if it is scheduled to transmit in more than one channel. Imaging the scenario the time slot is splittable and node A is already selected to transmit to node B using channel ch, denoting p_n as the probability that node A is also selected for transmission on n channels other than channel ch, then $\sum_n p_n \frac{n}{n+1}$ part of the time slot will be used to transmit in the channels other than channel ch. An example is given in the later part of this section to show how to calculate the probability p_n. The fraction of the time slot node A can use to transmit to node B in channel ch can be calculated as

$$p_{AB} = 1 - \sum_n p_n \frac{n}{n+1}, \quad (9)$$

To calculate the equivalent fraction, we consider two cases:

Case I: node A uses its LI to transmit data packets to node B: If node B's *LIC* is the same as node A's *LIC*, node A has to use its *LI* to transmit data packets to node B, as two interfaces of a node cannot be tuned to the same channel for transmitting and receiving at the same time. Since both nodes' *LI*s share the same channel, they will not be scheduled for receiving in the same time slot. If node B is scheduled for receiving and node A is scheduled for transmitting to node B, node A is always able to use its *LI* to transmit regardless of the channel usage of node A' *TI*. That is, node A can use all portion of the scheduled time slot, i.e., $p_{AB} = 1$.

Case II: node A uses its TI to transmit data packets to node B: To calculate the "equivalent" fraction, we first calculate the probability that node A is scheduled to transmit to other node(s) as well (We call it *Conflicting Probability*).

To calculate p_{AB} based on Equ. (9), we only need to analyze the case that node A is scheduled to transmit to node B and also scheduled to transmit over a channel other than B's *LIC* and A's *LIC*. Assume that node A has m downstream nodes which are assigned the same listening channel k, the probability of node A being scheduled to transmit on channel k is

$$p_{tch}(A \Rightarrow k) = \sum_{i=1}^{m} p_r(N_i^k) \cdot p_t(A \to N_i^k) \quad (10)$$

where N_i^k denotes the i^{th} downstream node of A with listening channel k. Functions $p_r()$ and $p_t()$ are calculated based on Eqs. (6) and (7) respectively.

We will use Fig.4 as an example to show how the *conflicting probability* is calculated. There are 4 channels, and A's *LIC* and B's *LIC* are channel 1 and 2 respectively, then we only need to calculate the probability of node A being scheduled for transmitting on channel 3 and 4 as $p_{tch}(A \Rightarrow 3)$ and $p_{tch}(A \Rightarrow 4)$ respectively based on Eq. (10). As only node C is assigned with channel 3, $p_r(C^{ch3}) = 1$. Assume A has the same opportunity of transmitting to C on channel 3 as node F, then $p_t(A \rightarrow C^{ch3}) = \frac{1}{2}$. Thus, $p_{tch}(A \Rightarrow 3) = p_r(C^{ch3}) \times p_t(A \rightarrow C^{ch3}) = \frac{1}{2}$. Similarly, assume D has the same chance of being scheduled in $ch4$ as node E and F, then $p_r(D^{ch4}) = \frac{1}{3}$. With $p_t(A \rightarrow D^{ch4}) = 1$, $p_{tch}(A \Rightarrow 4) = p_r(D^{ch4}) \times p_t(A \rightarrow D^{ch4}) = \frac{1}{3}$. As the scheduling in different channels is independent, we can calculate the probability of node A being scheduled in either channel 3 or 4 but not both, given that node A is already scheduled to node B, as

$$\begin{aligned} p_1 &= p_{tch}(A \Rightarrow 3)(1 - p_{tch}(A \Rightarrow 4)) \\ &\quad + p_{tch}(A \Rightarrow 4)(1 - p_{tch}(A \Rightarrow 3)) \\ &= \frac{1}{2} * (1 - \frac{1}{3}) + \frac{1}{3} * (1 - \frac{1}{2}) = \frac{1}{2} \end{aligned}$$

The probability of node A being scheduled in both channel 3 and 4 given that node A is scheduled to node B is

$$p_2 = p_{tch}(A \Rightarrow 3) p_{tch}(A \Rightarrow 4) = \frac{1}{2} * \frac{1}{3} = \frac{1}{6} \quad (11)$$

Assuming n take values 1 and 2, from Eq. (9), the "equivalent" fraction of the scheduled time slot A can really use to transmit to node B is

$$p_{AB} = 1 - \frac{1}{1+1} p_1 - \frac{2}{2+1} p_2 = 1 - \frac{1}{2} * \frac{1}{2} - \frac{2}{3} * \frac{1}{6} = \frac{23}{36} \quad (12)$$

That is, node A can only use 23/36 of the time slot scheduled for it to transmit to node B.

From the above example, we can see that a node will not waste any time slot if all its downstream nodes are in one channel. On the other hand, if a node has many downstream nodes assigned with many different channels, a larger fraction of time would be wasted. The transmission conflicting factor reflects the impact of interface constraint on network throughput.

Therefore, the delay factor on link AB due to conflicting schedule will be

$$f_c = \frac{1}{p_{AB}}, \quad (13)$$

which has a higher value if the fraction of the scheduled time slot a node can actually use is smaller.

Link Cost Calculation: By combining all the major delay factors mentioned above, the link cost for AB is defined as

$$\begin{aligned} W_l &= f_t \cdot f_r \cdot f_s \cdot f_c \\ &= \frac{1}{W} \cdot \frac{1}{1-p} \cdot \left(\frac{1}{p_r(B)} \cdot \frac{1}{p_t(A \rightarrow B)}\right) \cdot \frac{1}{p_{AB}}. \quad (14) \end{aligned}$$

From the cost analysis discussed above, to calculate the cost of an incoming link of a node, the cost factors f_s and f_c can be calculated based on the network topology and existing channel assignments for the nodes within an interference range.

Eq. (14) can be understood in an intuitive way. Given the link from node A to node B, for one unit of time, node B can be scheduled as a receiver for $p_r(B)$ time unit, whose $p_t(A \rightarrow B)$ part will be assigned to the link between A and B. Within that fraction of the time unit, node A uses only p_{AB} portion to transmit to node B at the rate of W and needs $\frac{1}{1-p}$ transmissions for each packet. Therefore, the total link delay will be $O(\frac{1}{W \cdot (1-p) \cdot P_r(B) \cdot p_t(A \rightarrow B) \cdot p_{AB}})$. As $f_r = \frac{1}{1-p}$ is a constant, it can be ignored during path searching.

B. Channel Assignment and Path Setup

Based on the link cost model, we propose an on-demand routing protocol. With multiple interfaces, initially, each node picks one interface as its *LI* and then randomly selects a channel to tune the *LI* to. If a source node needs a path to the destination, it broadcast a Route Request (RREQ) packet to its one-hop neighbors by sending the message to all the available channels. When a node i receives a RREQ packet, it will generate an updated RREQ packet to broadcast if necessary. The updated RREQ packet carries the *accumulative cost* of the minimum cost subpath from the source to node i, the (ID, Assigned LIC) pairs for nodes along the subpath, the capacity of node i's *TI*, and for each downstream node j, the number of nodes sharing the same *LIC* as j and within its interference range.

Once a node receives a RREQ packet, it will extend the subpath indicated in the RREQ packet to itself. If the node already has an *LIC* assigned when setting up other paths, it simply calculates the new accumulative subpath cost based on its *LIC*. Note that we don't assume a centralized scheme exist to assign the channels for all the paths at the same time. Channels assigned during previous path setup will not be modified during new path setup. Channel assigned to a node can be modified during route maintenance as discussed in Section V-C or when a path is refreshed to track the updated network topology. If the node has not been assigned a listening channel, it needs to calculate the minimum cost for the subpath by inspecting every possible channel assignment for its *LI*, and notes down the channel that provides the minimum cost as a candidate *LIC*. The node then broadcast a new RREQ packet.

Given a channel ch, the cost of the link between the sender A and the receiver B can be calculated using Eq. (14) after determining the four major factors as follows.

1) Interface Capacity factor: The receiver will determine the common rate (W) supported by the two interfaces of the sender and the receiver.
2) Retransmission factor: As our scheduling algorithm constrains the load of a channel in a time slot, f_r is very small and thus not considered during path searching to avoid the difficulty in measuring conditions of multiple channels.
3) Channel and scheduling factor: The receiver B first checks the number of nodes within its two-hop neighborhood using ch as *LIC* ($N_{B,ch}^{2-hop}$) and the number of its upstream nodes (N_{ToB}). Both values could be changed after the path is set up, so the change should be taken into account in advance. If A is not an upstream node of node B yet, after the path is set up, N_{ToB} should be increased by 1. $N_{B,ch}^{2-hop}$ also needs to be adjusted based on the channel assignment for previous hops. Denoting

the list of node entries included in the RREQ packet as $nodelist$ and B's two-hop neighbors as N_B^{2-hop}, the adjusted $N_{B,ch}^{2-hop}$ can be calculated using Algorithm 3, where $N_{B,ch}^{2-hop}$ will be adjusted if the relationship between "to be assigned channel" ($channel$) for node n carried in the $nodelist$ and the possible channel assignment (ch) for B has changed. Once both information is obtained, then node B can calculate f_s based on Eq. (8).

Algorithm 3 $AdjustedContendingNum(nodelist, ch)$

for all node $n \in nodelist$ **do**
  if (n.NodeID $\in N_B^{2-hop}$) **then**
    if (n doesn't have assigned LIC $\wedge$ n.channel=ch) **then**
      $N_{B,ch}^{2-hop} \leftarrow N_{B,ch}^{2-hop} + 1$; {The contending from n is not counted by N_{ch}^{2-hop} now, and needs to be counted when *n'LI* is committed to ch after path establishment}
    end if
  end if
end for
return $N_{B,ch}^{2-hop}$

4) Conflicting factor: The sender includes all necessary information in the RREQ packet for the receiver to calculate f_c based on Eq.(9).

A receiving node will not tune its *LI* to the assigned channel immediately, but will wait until the path is confirmed by the destination. When the destination receives a RREQ packet, it can respond with a RREP immediately to confirm the new path if the total path cost is smaller than the one recorded, or it can wait for some interval of time and only respond to the RREQ which finds the minimum cost path within the interval. The latter option would reduce the control overhead at the cost of a higher route setup delay. Once receiving a RREP packet, a node will tune its *LI* to the assigned *LIC* if the assignment is new, and notify its neighbors through a *Channel Update Message*.

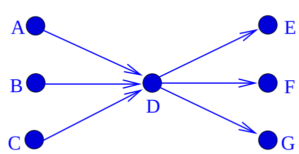

Fig. 5. An Example for Channel Assignment and Path Setup.

An example is shown in Fig. 5 to explain how our channel assignment and path setup work. Assume the data rate for each link is the same, so the interface capacity factor (f_t) is constant and the same for all links. For the convenience of presentation, we assume $f_t f_r$ equals 1 and there are two channels in the network. Initially, no node is assigned a listening channel. First, source node A broadcast a RREQ message to search for a path to destination D. After receiving the RREQ message, node D calculates the cost of link AD by examining the use of channel 1 and 2 respectively. Since no other nodes have been assigned channels yet, the link cost is 1 for both channel 1 and 2, and thus node D can pick either one as the *to be assigned channel* (before it is confirmed by the destination). Here we assume channel 1 is selected as indicated in Table I. Then D rebroadcasts the RREQ packet and node G receives it. Knowing from RREQ the *to be assigned channel* for node D is channel 1, node G determines the link cost for link DG to be 2 when channel 1 is used and 1 when channel 2 is used. So node G will choose channel 2 and the total cost for path ADG is $1+1=2$. Since this path cost is the minimum, path ADG will be selected and node D and G will be assigned channel 1 and 2 respectively. We then look at the path searching for source node B and destination node F. Since node D is already assigned channel during the path setup for ADG, it will keep the assignment. Assume that B and A have the same chance of transmitting to D, the cost for link BD is thus 2. After F receives the RREQ from node D, it calculates the link cost for DF, which are 4 (i.e., $f_s = 2$, $f_c = 2$) and 2 corresponding to channel 1 and 2 respectively. F will then be assigned channel 2. Similarly, the channel assignment for node E is 2 and the path for source node C and destination node E is CDE as shown in Table I. It is worth mentioning that the channel assignment and path searching in this example leads to minimum cost paths. The data flow from nodes A, B, and C to D will not affect the data flow from D to nodes E, F, and G.

C. Route Maintenance

Due to the environmental changes or mobility, the path found in the route discovery phase may no longer be as efficient. To ensure consistent performance, our routing algorithm includes a route maintenance scheme to adapt the path and channel assignment based on the changes of topology, traffic, and channel condition.

1) Channel Switching: A node is updated with the channel assignment of all its two-hop neighbors periodically. We consider three channel switching scenarios: a) *Balancing load among channels*. If a node finds it has a lot of queued data for a receiver, it can notify the receiver to switch to a channel with fewer sharing neighbors. To make sure that the channel change will not increase the delay of the overloaded path, the receiver will check the cost of the path segment passing through itself and within its two-hop range. Suppose node C on a path (A $\rightarrow$ B $\rightarrow$ C $\rightarrow$ D $\rightarrow$ E $\rightarrow$ F $\rightarrow$ G) finds it has long queued data for D, D needs to check if it can switch its *LI* to a new channel by comparing the total link cost of the segment BCDEF using the new channel and using the existing channel. It can switch to the new channel if the channel change does not increase the cost of its path. b) *Improving performance around a "hot" node.* If several paths pass through a node (X), i.e., a busy node, node X can check if changing to a different channel would lead to the cost reduction in some paths while not increasing the cost for the remaining paths. If so, it will switch to the new channel. c) *Avoiding the channel with high error rate.* As our scheduling algorithm constrains the number of nodes competing in a channel, the collision probability will not be high. If the measured packet loss rate is very high (partially due to errors), then the channel will be changed. The switching of channel to balance the channel and interface usage in a neighborhood, also helps to improve fairness among neighboring nodes.

2) Replace Operation: If a node has either a *TI bottleneck* or *LI bottleneck*, it will look for an alternative path that goes through a *replacement node* to forward the data. The replacement node should make sure the new path passing through itself will not have a higher end-to-end delay than the old path. Given a path segment (A $\rightarrow$ B $\rightarrow$ C $\rightarrow$ D $\rightarrow$ E), if C has an interface bottleneck, C will check the path passing through a neighboring node within B's and D's transmission range, say a node F. Node

TABLE I
LINK COST AND PATH COST

Channel No	AD cost	DG cost	ADG cost	BD cost	DF cost	BDF cost	CD cost	DE cost	CDE cost
ch1	**1**	2	2	**2**	4	4	**3**	4	6
ch2	1	**1**			**2**			**3**	

C will compare the total cost for (A → B → F → D → E) with the cost of the current path segment. If the new cost is smaller, node C will send the message to node B, F and D to notify the path change so that node B will send the packets to node F which will forward the packets to node D.

3) Remove operation: Given a path segment (A → B → C), if node A detects that both B and C are its one-hop neighbors, it can forward the data packets to node C directly.

4) Insert Operation: Given a path segment (A → B), if the signal received from A is less than some threshold, node B will broadcast a request in its neighborhood. If node C can reach both A and B and can receive signals from both at good quality, it can insert itself between node A and node B.

To reduce implementation cost, the above maintenance schemes are only based on local information. However, our performance studies in the next section demonstrate that our schemes can effectively maintain the network throughput in mobility scenario.

VI. PERFORMANCE EVALUATION

We implemented our proposed algorithms using simulation package Glomosim [27]. Each node is assumed to have only two 802.11a interfaces, with interface rate 54 Mb/s. The time slot length is set to $10ms$ (about 35 maximum-length packet transmission time [11]), the broadcast interval of Hello Messages is set to $5s$, and the backoff window sizes for $W0$,$W1$,and $W2$ in the Prioritized Transmitting scheme (Sec. IV) are set to 7, 15, and 31 respectively. The transmission power is $15dBm$, the radio sensitivity is $-84dBm$, and the radio receiving threshold is $-74dBm$. We compare the performance using our integrated MAC and routing framework with scheme using independent MAC and Routing (e.g., DCA (Dynamic Channel Assignment) [9] as MAC and AODV as routing) as well as the scheme simply using AODV over IEEE 802.11a. One reason of selecting DCA is because it also uses two interfaces, which can provide more fair comparison as compared to schemes using only single interface or the ones using the number of interfaces larger than two. In DCA scheme, one of the channels is used as control channel while the remaining channels are used for data transmissions. Each node uses one interface to monitor and transmit on the control channel, and the other interfaces to transmit and receive data packets on data channels. Before each transmission, two nodes exchange information in control channel to select a channel to transmit data. Then, the sender broadcast a RES message over control channel to reserve the data channel and sends the data packet to the receiver.

CBR is used as the application protocol. To provide enough traffic load to study the multi-channel benefit, the size of packet is set as 2000 bytes and packets are sent out every 0.5 ms. Each simulation runs 100 seconds. For each run, we try to get the maximum throughput by tuning CBR rate and hence the network load. Each simulation result is obtained by averaging over multiple runs with different random seeds. We evaluate the performance with use of 2, 3, 4 and 5 orthogonal channels respectively. For the rest of this section, we use ‘Joint-x’, ‘DCA-x’ (x is the number of channels) and ‘802.11’ to represent our scheme, the AODV over DCA scheme, and the AODV over 802.11a scheme respectively.

A. *Chain-Topology*

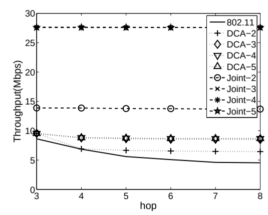

Fig. 6. Throughput in chain topology

We first evaluate our protocol over a simple chain topology with 9 nodes. Only one CBR flow is set up from node 0 to one of the last 6 nodes (i.e., the hop count of the flow will be from 3 to 8 hops). The simulation results are shown in Fig. 6. It is obvious that our protocol performs much better than DCA scheme and 802.11.

If there are only two channels, similar to 802.11, DCA can only use one channel for data transmission. However, by separating control channel and data channels, the control packet collision and hence the number of retransmissions in DCA can be reduced. Therefore, DCA performs a little bit better than 802.11. With more available channels, the number of data channels that DCA can use increases. When having three channels, one channel (e.g. 3) will be used as control channel, the remaining two will be used as data channels. In a snapshot of the network, the best channel assignment for the links along the chain could be like “..., channel 1, idle, channel 2, idle, channel 1, idle, ...”. The link between two active links is kept idle as a DCA node only has one interface available for data transmission, and links within two-hops cannot be assigned the same channel to avoid interference. Adding the third data channel cannot improve throughput. This is the reason why the curves of DCA-3, DCA-4 and DCA-5 overlap in Fig. 6.

In contrast, our protocol can make a better use of more channels. If there are only 2 channels, in a network snapshot, the best channel usage for the links along the chain could be like “..., channel 1, channel 2, idle, channel 1, channel 2, idle, ...”. With 3 channels, our protocol could achieve better throughput. The network snapshot could be like “..., channel 1, channel 2, channel 3, channel 1, channel 2, channel 3, ...”, i.e. all the links are active in transmitting and three channels are enough to obtain the maximum throughput in the chain topology. Therefore, the curves of Joint-3, Joint-4 and Joint-5 overlap in Fig. 6.

B. *Grid Topology*

In this simulation, we evaluate the performance of our protocol in a more practical scenario, i.e., a 5X5 grid network. The grid distance is set that the receiving power at a neigbhoring node is -70dBm. We set up four CBR connections as shown in Fig. 7(a). These four CBR connections will make the center of the grid more congested. The simulation results for aggregate network throughput are shown in Fig. 7(b).

The throughput of DCA improves significantly when the number of channels is increased from two to three, but the rate of improvement reduces with further increase of the number of channels, as the routing protocol cannot take advantage of multiple channels to build efficient paths. While for our protocol, as compared to 802.11, the throughput increases almost linearly with the number of channels. With integrated routing and MAC design, our protocol can utilize multi-channel resources very efficiently, and our scheduling scheme effectively mitigates the limitation in the number of interfaces.

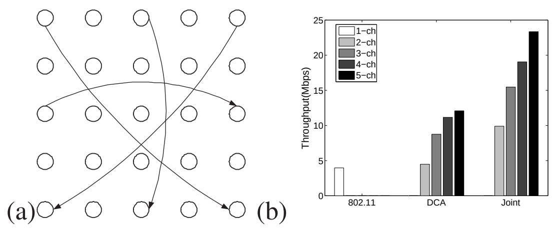

Fig. 7. Performance for grid topology: a) Topology; b) Throughput.

C. Random Topology

In this set of simulations, nodes can move randomly within a $1000m \times 1000m$ network area. The movement follows the improved random waypoint model [28]. As we use 802.11a which has a lower transmission range than 802.11 b, the default average moving speed is set to $5m/s$, and maximum speed is set to $10m/s$. A connection is established by randomly picking a source and a destination. We study the impact on performance of load, node density and mobility.

We first study the impact of traffic load. There are total 50 nodes in the simulated network area, and the number of CBR connections is varied from 6 to 10. In Fig. 8(a), we can see that the total throughputs of our protocol under different number of channels are much higher than those using other schemes. The aggregate throughputs for both 802.11 and DCA-2 (having one data channel) decrease as the number of connections increases. This is because adding connections to an already saturated network area will introduce more collisions and lead to throughput degradation. When the number of channels increases, the saturation gets released, but the throughput increase for DCA is small as the routing protocol could not take advantage of multiple channels to build efficient paths to support more connections. For our protocol, the throughput of Joint-2 increases slightly because the network is saturated with only 2 channels. With more channels, the throughput of our protocol has a larger increase at a higher load as compared to DCA because our protocol can handle additional connections more efficiently by routing the traffic away from the saturated area and assigning channels based on the traffic.

To evaluate the impact of node density, we have eight CBR connections in the network and vary the number of nodes from 40 to 60. The simulation results in Fig. 8(b) again show that our protocol can achieve much higher throughput increase as node density increases, while the aggregate throughputs of 802.11 and DCA-2 reduce slightly and the throughput of DCA remains almost constant when more channels are used. The trends are similar to those from the study of load impact. When the node density increases, the network load will also increase with a higher contention in a network area. However, our protocol can better take advantage of available nodes and radio interfaces to build more efficient routing path, and route traffic away from bottlenecks during route maintenance.

Last, we study the impact of mobility on the protocols. There are eight CBR connections in the network and the number of nodes is 40. The average speed is varied from $4m/s$ to $20m/s$. The simulation results for aggregate throughput are shown in Fig. 8(c). As expected, the throughput for all three protocols decrease when the speed increases as a result of the link breakage during mobility. In addition, the decrease is faster when more channels are used. As the average link throughput will increase with a higher number of channels, a link breakage will have a higher impact on the throughput. However, the throughput of our protocol remains much higher than DCA in different mobility cases and the throughput reduces much slower than the reference schemes, which indicate that our maintenance scheme can adapt the path and channel assignment effectively to topology changes thus preventing link breakage in advance.

VII. Conclusion

In this paper, we propose an integrated MAC and routing design to explore the capabilities provided by multiple channels and multiple interfaces in ad hoc networks. We define a new routing metric that considers the difference in interface speeds, the delay due to retransmission, the impact of interface constraint, and the delay due to node competition for the limited number of channels. Based on the routing metric, we propose a routing algorithm for path discovery that considers all the major factors of a MCMI network in finding the minimum cost path. We also present route maintenance schemes to adapt the path and channel setup in the face of network dynamics. Given the channels assigned during path setup, our scheduling scheme explores the resources at time domain to coordinate channel usage and interface sharing among neighboring nodes to constrain the number of competing senders in a time slot thus reducing interference in a channel. The scheduling also helps to minimize the effect of channel switching delay, balance the load and enable fairness among neighboring nodes. Additionally, we enhance the 802.11 MAC with prioritized transmission to resolve collisions among nodes scheduled to transmit on the same channel in the same time slot, reduce the broadcast delay in an MCMI environment, and allow nodes to opportunistically use the spare channel resources to further improve the throughput. Simulation results demonstrate that our integrated framework can utilize the channel resources very efficiently to significantly improve the network throughput in a multi-channel multi-interface environment.

References

[1] A. Raniwala, K. Gopalan, and T. Chiueh, "Centralized algorithms for multi-channel wireless mesh networks," *ACM Mobile Computing and Communications Review*, Apr 2004.

[2] A. Raniwala and T. Chiueh, "Algorithms for an IEEE 802.11-based multi-channel wireless mesh network," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, Mar 2005.

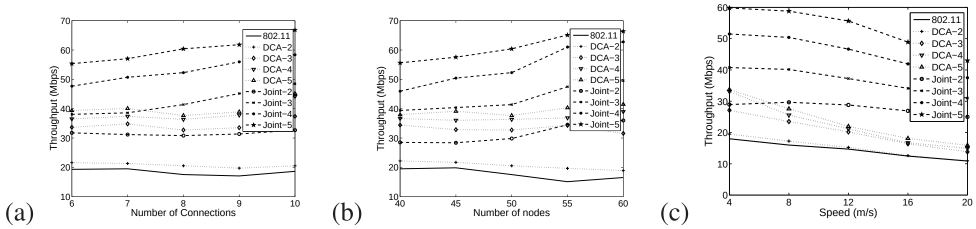

Fig. 8. Throughput for random topology: a) Effect of load; b) Effect of node density; c) Effect of mobility.

[3] M. Alicherry, R. Bhatia, and L. Li, "Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks," in *Proceedings of ACM MobiCom*, Sept. 2005.

[4] A. Mishra, D. Agrawal, V. Shrivastava, S. Banerjee, and S. Ganguly, "Distributed channel management in uncoordinated wireless environments," in *ACM Mobicom 2006*, Los Angeles, CA, Sept. 2006.

[5] K. N. Ramachandran, E. M. B. and K. C. Almeroth, and M. M. Buddhikot, "Interference-aware channel assignment in multi-radio wireless mesh networks," in *IEEE INFOCOM 2006*, Apr 2006.

[6] M. Kodialam and T. Nandagopal, "Characterizing the capacity region in multi-radio, multi-channel wireless mesh networks," in *Proceedings of ACM MobiCom*, Sept. 2005.

[7] A. Nasipuri and S. R. Das, "Multichannel CSMA with signal power-based channel selection for multihop wireless networks," in *IEEE Vehicular Technology Conference (VTC)*, Sept. 2000.

[8] N. Jain and S. R. Das, "Protocol with receiver-based channel selection for multihop wireless networks," in *IEEE International Conference on Computer Communications and Networks (IC^{3}N)*, Oct 2001.

[9] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, and J.-P. Sheu, "New multi-channel MAC protocol with on-demand channel assignment for mobile Ad Hoc networks," in *International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN)*, Oct 2000, pp. 232–237.

[10] J. So and N. H. Vaidya, "Multi-channel MAC for Ad Hoc networks: Handling multi-channel hidden terminals using a single transceiver," in *ACM International Symposium on Mobile Ad Hoc Networking and Computing*, May 2004.

[11] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks," in *Proceedings of ACM MobiCom*, Sept. 2004.

[12] A. Tzamaloukas and J.J.Garcia-Luna-Aceves, "A receiver-initiated collision-avoidance protocol for multi-channel networks," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, Apr 2001.

[13] J. So and N. H. Vaidya, "A routing protocol for utilizing multiple channels in multi-hop wireless networks with a single transceiver," in *Technical report*, Oct 2004.

[14] P. Kyasanur and N. Vaidya, "Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks," *SIGMOBILE Mobile Computing and Communications Review*, vol. 10, no. 1, pp. 31–43, Jan 2006.

[15] X. Lin and S. Rasool, "Distributed and provably-efficient algorithms for joint channel-assignment, scheduling and routing in multi-channel ad hoc wireless networks," accepted to IEEE/ACM Transactions on Networking.

[16] S. Merlin, N. H. Vaidya, and M. Zorzi, "Resource allocation in multi-radio multi-channel multi-hop wireless networks," in *IEEE INFOCOM*, Apr 2008.

[17] J. Shi, T. Salonidis, and E. W. Knightly, "Starvation mitigation through multi-channel coordination in csma multi-hop wireless networks," in *MobiHoc '06: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*. New York, NY, USA: ACM, 2006, pp. 214–225.

[18] D. B. Johnson, D. A. Maltz, and Y.-C. Hu, "The dynamic source routing protocol for mobile ad hoc networks (DSR)," in *IETF MANET Working Group (Draft 10)*, 2004.

[19] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," in *IETF RFC 3561*, Jul 2003.

[20] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *ACM Mobicom*, Sept. 2004.

[21] W.-H. Tarn and Y. C. Tseng, "Joint multi-channel link layer and multi-path routing design for wireless mesh networks," in *IEEE INFOCOM*, May 2007.

[22] D. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *ACM Mobicom*, Sept. 2003.

[23] H. Li, Y. Cheng, C. Zhou, and W. Zhuang, "Minimizing end-to-end delay; a novel routing metric for multi-radio wireless mesh networks," in *IEEE INFOCOM*, Apr 2009.

[24] L. Bao and J. Garcia-Luna-Aceves, "Hybrid channel access scheduling in ad hoc networks," in *IEEE ICNP*, Nov 2002.

[25] J. Padhye, S. Agarwal, V. N. Padmanabhan, L. Qiu, A. Rao, and B. Zill, "Estimation of link intreference in static multi-hop wireless networks," in *Proceedings of Internet Measurement Conference (IMC)*, Oct 2005.

[26] G. Bianchi and I. Tinnirello, "Kalman filter estimation of the number of competing terminals in an ieee 802.11 network," in *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, Mar 2003.

[27] X. Zeng, R. Bagrodia, and M. Gerla, "GLOMOSIM: a library for parallel simulation of large-scale wireless networks," in *Proceedings of the 12th Workshop on Parallel and Distributed Simulations (PADS)*, May 1998.

[28] W. Navidi and T. Camp, "Stationary distributions for the randomwaypoint mobility model," *IEEE Trans. on Mobile Computing*, vol. 3, no. 1, pp. 99–108, 2004.

PLACE
PHOTO
HERE

Jinhua Zhu received his Ph.D. degree in Computer Science and Engineering from University at Buffalo in 2005. He is currently a service engineer in Ericsson Inc. His research interests include wireless sensor networks and cognitive radio.

Xin Wang (M'01/ACM'04) received her Ph.D. degree in electrical and computer engineering from Columbia University, New York, NY.

She is currently an Assistant Professor in the Department of Electrical and Computer Engineering of the State University of New York at Stony Brook, Stony Brook, NY. Before joining Stony Brook, she was a Member of Technical Staff in Bell Labs Research, Lucent Technologies, New Jersey, and an Assistant Professor in the Department of Computer Science and Engineering of the State University of New York at Buffalo, Buffalo, NY. Her research interests include wireless networks and communications, mobile and distributed computing. Dr. Wang has served in numerous technical communities of top networking conferences.

Dahai Xu (S'01-M'05) is currently a research staff in AT&T Labs. After receiving his Ph.D. degree from University at Buffalo in 2005, he spent two years as a postdoctoral research associate in Princeton University. His research interests include Internet design, control and management; Algorithm design and fast implementation; Large-scale non-linear network optimization; Secure communication in wireless ad hoc networks.