A Unified MAC and Routing Framework for Multichannel Multi-interface Ad Hoc Networks

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

***Abstract*—Improving the capacity of wireless networks is critical and challenging. Although wireless standards such as IEEE 802.11 allow the use of multiple channels at the physical layer, current Media Access Control (MAC) and routing protocols of mobile ad hoc networks have mainly been developed to run 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, as well as feedback from the MAC layer. Given the routing path and channel assignment, our scheduling scheme at the MAC layer explores the resources at the time domain to coordinate transmissions within an interference range to maximize channel usage, reduce channel access competition among nodes assigned to 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 to reduce the transmission delay of mission-critical packets and message broadcast, which help 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 multichannel multi-interface ad hoc networks.**

***Index Terms*—Ad hoc networks, cross layer, Media Access Control (MAC), multichannel, multiradio, routing.**

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 local area network (LAN) standards such as IEEE 802.11 often allow for transmissions on multiple physical channels, current Media Access Control (MAC) and routing protocols in infrastructure-free ad hoc networks are generally designed to transmit data only on one channel. In addition, 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 that 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 paper 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 multichannel multi-interface (MCMI) environment. Because 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 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. Aside from these issues, in a multihop network, it is critical and challenging to establish a routing path that exploits the MCMI feature for better throughput and to 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 multichannel environment.

Because the aforementioned 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, i.e., to exploit MCMI capabilities in mobile ad hoc networks to fully use the available spectrum to improve the network performance. Our framework jointly considers routing and 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

Manuscript received October 28, 2009; revised January 11, 2010 and June 12, 2010; accepted September 5, 2010. This work was supported in part by the U.S. National Science Foundation under Grant CNS 0751121 and Grant CNS 0628093. The review of this paper was coordinated by Prof. A. Boukerche.

J. Zhu is with Ericsson Inc., Boulder, CO 80301 USA.

X. Wang is with the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794 USA.

D. Xu is with the AT&T Labs Research, Florham Park, NJ 07932 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2010.2080362

the path. Our route-maintenance scheme adopts the path and channel assignment based on changes of topology and channel condition and on feedbacks from the MAC layer. Given the channel assignments during path setup, a *scheduling scheme* is used at the MAC layer to coordinate the channel usage and interface sharing/switching to enable communications between nodes and 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 interfaces is smaller than the number of available channels. Our contributions can be summarized as follows:

- Design 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.
- Develop 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.
- Incorporate 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.
- Design a scheduling scheme that manages resources in the time dimension to coordinate channel usage and interface sharing among neighboring nodes assigned the same channel to reduce channel competitions, to 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.
- Enhance 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 multichannel broadcast delay and the transmission delay for mission critical applications, and allow unscheduled nodes to opportunistically use the available channel resources to improve throughput.

Multichannel multiradio wireless networks have received a substantial amount of recent interest, particularly 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]. Because 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, whereas using single interface [13] in multichannel environment for routing would result in poor connectivity.

To the best of our knowledge, this paper provides the first practical network framework that concurrently considers routing and channel assignment at the 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 to avoid the deafness problem when a transmission pair tunes their radio interfaces to the same channel at different times. 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 the 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 the delay of mission-critical traffic and control messages.

The rest of this 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 that pertain to a MCMI 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 detail a joint routing and channel assignment scheme and an efficient channel and route-maintenance scheme. Section VI describes our evaluation using simulations. We conclude this 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, whereas 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] and [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] and [11], the authors propose multiple access schemes for the nodes equipped with single interface. Receiver-initiated channel-hopping with dual polling (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, whereas the scheme in [17] targets addressing the starvation problem in a Carrier Sense Multiple Access (CSMA)-based multihop wireless network.

Predominant routing protocols such as dynamic source routing (DSR) [18] and ad hoc on-demand distance vector (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, because 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 interdependent and that transmission on the same path may experience intrachannel 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 apply in the mobile ad hoc networks, which require a distributed scheme to quickly react to topology change. The scheme proposed in [21] combines multichannel link layer with multipath routing. Although interesting, many design ideas [e.g., superframe pattern, dynamic adjustment of the transmit–receive (T/R) ratio, and multipath routing] proposed in this paper target to address the inefficiency due to the half-duplex transmissions as a result of using one radio interface at each node. The use of a single interface would lead to more severe multichannel hidden terminal problem [10] and deafness problem. In [20], the authors extend the work in [22] and propose a new routing metric, i.e., weighted cumulative expected transmission time (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 intrapath 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 that the channel has been assigned, the work in [23] considers queuing 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 the 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 for implementing the algorithms. Although the proposed scheme is not centralized, a supernode 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 neighborhoods 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 an MCMI network. The scheduling algorithm is purely distributed, and each node can make a scheduling decision to efficiently coordinate channel usage and interface switching with no need for complicated signaling messages.

III. System Overview

The goal of this paper 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 the *frequency* domain through channel assignment and the *time* domain through transmission time slot scheduling to significantly increase the network throughput. Our design at the routing layer includes the following techniques: 1) a *link cost model* for capturing the characteristics of MCMI networks and the impact of MAC-layer scheduling; 2) a *joint channel assignment and routing scheme* for concurrently searching for the minimum cost path and assigning channels to nodes along the path; and 3) a route-maintenance scheme for adapting the path and channel assignment in response to changes of network topology and channel conditions and MAC feedback. Given channels assigned during the path setup, our design at the MAC layer includes the following techniques: 1) a *distributed scheduling scheme* for coordinating the channel usage in the unit of time slot to reduce competition among nodes assigned the same channel within an interference range and for coordinating interface sharing and switching to reduce transmission conflict and unnecessary switching delay and 2) a *prioritized transmission scheme* for coordinating 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 multichannel 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 the following two types: 1) the *listening interface* (*LI*) and 2) the *transmitting interface* (*TI*). During path setup, one radio interface of a node will be designated as *LI* and assigned a channel, called the *LI channel* (*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 TIs, which can flexibly be 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 periodically be sent by a node to maintain network topology, as is 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 quickly reduces as the distance between the transmitter and receiver increases. To reduce the implementation overhead, in this paper, we consider interference of 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*.

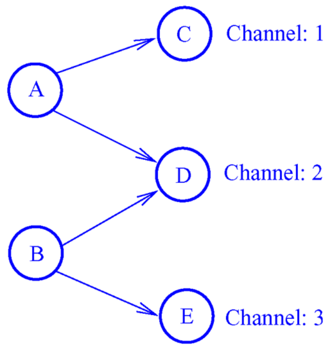


Fig. 1. Example of transmission coordination.

IV. Media Access Control 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 the delay of important control and data packets. Our MAC scheme addresses the following issues.

1) *Interference among the transmissions over the same channel*. There are generally a limited number of channels in the system. Due to cost, time, and policy constraints, the number of channels to which a node can tune 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 the use of multiple channels, an interface needs to be switched among different channels. Because channel switching incurs a nonignorable delay [11], it would be more efficient to reduce channel switching.
3) *Transmission conflict*. A node may have several downstream nodes that listen to different channels. With no coordination, independent transmissions from multiple upstream nodes to the same channel will result in collisions, whereas better channel usage coordination would lead to concurrent transmissions. For example, in Fig. 1, node A can transmit to nodes C and D using channels 1 and 2, respectively, whereas node B can transmit to nodes D and E using channels 2 and 3, respectively. Without any coordination, nodes A and B may try to transmit to node D using channel 2 at the same time, whereas neither channel 1 nor channel 3 is used, which causes both collision at the same receiver and channel resource wastage.
4) *Broadcast delay*. Because 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 condition 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).

A. Channel-Scheduling Scheme

In a MCMI system, a simple exchange of request to send/clear to send (RTS/CTS) between a sender and a receiver on the LIC of the receiver is not enough to avoid the hidden terminal problem, because a potential interference node may be listening to a different channel, whereas 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 that 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 LIC, 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; and 3) when a node is scheduled to transmit to multiple receivers with different LICs, 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-A2, 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 (see 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 significant throughput degradation under heavy load as in a pure CSMA with collision avoidance (CSMA/CA)-based scheme such as IEEE 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. Because 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 of 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. The data packet from the sender to the receiver is generally longer than the confirmation packet from the receiver to the sender; therefore, it is more important to reduce interference at the receiver side. Our scheduling has the following two phases: 1) *receiver scheduling* and 2) *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 the 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 *with no need for 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 has the highest priority, the entity with the largest ID will be selected.

This algorithm is summarized in 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 that contend for the same *LIC* (ch) as i. $Rand(X,t)$ is adopted from the $Hash()$ function used in [24]. We have

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

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 that the size of the output of $Hash()$ function is 64 b. 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.

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
```

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 of its two-hop neighbors that 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 that one or more nodes are scheduled for receiving, node S will check

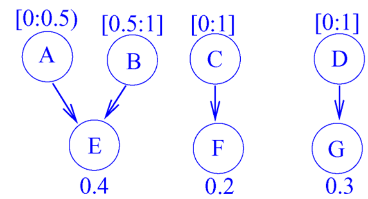


Fig. 2. Example of scheduling.

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 a nonoverlapping *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
```

One example is shown in Fig. 2 to explain how our scheduling works. There are four senders (nodes A, B, C, and D) and three receivers (nodes 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 (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. Because 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. Because node E's random value (0.4) falls within node A's probability range, i.e., [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 node B is not scheduled to transmit to node E. Nodes C and D determine that nodes 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

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

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

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

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 a 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 frequently happen, because 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 the 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 is 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, which will be discussed as follows.

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-A2). With the support of time-slot-based scheduling, the following additional issues should still be addressed.

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., route request (RREQ), route error (RRER), and route reply (RREP) packets] and mission-critical data packets. To avoid a collision in transmitting the first priority packets, each node waits for some random time within a window $W0$.

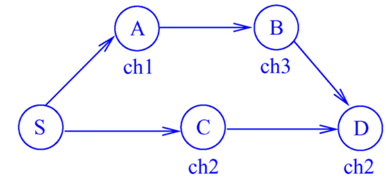


Fig. 3. Two possible paths.

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 an RTS packet but with a different delay window $W1$ larger than $W0$.

The *third* level of priority will be assigned to the nonscheduled senders to avoid wasting the time slots that cannot be used up by the scheduled transmissions. To avoid competing with the scheduled sender, a nonscheduled 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 nonscheduled nodes only need to wait for a random period of time within the window $W2$ before transmitting subsequent packets. In addition, a nonscheduled 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.

Note that our scheme is robust in the 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 the 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 nonscheduled nodes with a lower priority. We will show in the next section that more than one node within a 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 nodes C and D in Fig. 3. Assume that each link has the same transmission rate. Although path SCD has only two hops, because nodes C and D are assigned the same LIC ($ch2$), 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 nodes 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 paper, 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.

A routing protocol generally searches for the minimum cost path between the source and the destination. Because 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 nonlinear 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 suboptimal 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 the 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., 11Mb/s in IEEE 802.11b and 54Mb/s 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, because 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. To measure the condition of a channel, there is also a need to first transmit data on the channel, which may not be possible before the channel is assigned. The interference measurement in [25] can be only used for static networks. Instead, we analytically estimate the packet error rate based on our scheduling scheme.

Assume that the interference range is about twice the transmission range. In our scheduling scheme, only one receiver is scheduled within a two-hop neighborhood. Assuming that the network area is A, the transmission range is R, and the nodes are evenly distributed. If the scheduled receivers are at 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

$$N_s^{2-hop} = (N_s/A) \cdot \left(\pi(2R)^2\right) = 4 \qquad (3)$$

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 IEEE 802.11, as described in Section IV-B. Most transmission failures are due to collisions (e. g., collisions in RTS messages). For an IEEE 802.11 network, the collision probability or packet error rate p is impacted by the number of contending nodes n [26], i.e.,

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

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

Because our scheduling algorithm restricts the average number of competing nodes within the interference range to be a constant number 4, based on (4), the average packet error rate p is small and a constant. The expected number of transmissions (*ETX*) can be calculated as $1/(1-p)$. 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) as follows:

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

Because 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. Although scheduling helps 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 that share the same *LIC* within a two-hop neighborhood, only one node is scheduled for receiving in a slot. Assuming that each node has the same probability of being scheduled for receiving and node B is assigned channel ch as its *LIC*, the probability that node B is scheduled for receiving in channel ch is

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

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

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

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

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

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

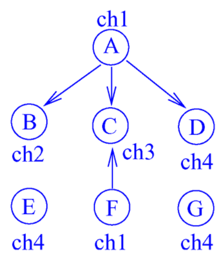


Fig. 4. Transmission-conflicting example.

This factor reflects the impact of network topology and channel constraint on the network throughput. If there are a large number of nodes that share 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 the 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, because scheduling is distributedly performed in reference to each receiver, it may be scheduled to transmit 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 channels 2, 3, and 4, respectively. Node A is also scheduled to transmit to all the three nodes. Because 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}, i.e., 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.

The concept of equivalent fraction of the time slot can be understood in an intuitive way. We assume that 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. Suppose that 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, $\sum_n (p_n n/n+1)$ part of the time slot will be used to transmit in the channels other than channel ch. One example is given in the latter part of this section to show how we can calculate the probability p_n. The fraction of the time slot that node A can use to transmit to node B in channel ch can be calculated as

$$p_{AB} = 1 - \sum_n p_n \frac{n}{n+1}. \tag{9}$$

To calculate the equivalent fraction, we consider two cases:

Case 1: 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, because two interfaces of a node cannot be tuned to the same channel for transmitting and receiving at the same time. Because both nodes' *LIs* 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 can always use its *LI* to transmit, regardless of the channel usage of node A's *TI*. That is, node A can use all portions of the scheduled time slot, i.e., $p_{AB} = 1$.

Case 2: 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 also scheduled to transmit to other nodes (we call it *conflicting probability*).

To calculate p_{AB} based on (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*. Assuming that node A has m downstream nodes, which are assigned the same LIC k, the probability that node A is scheduled to transmit on channel k is

$$p_{tch}(A \Rightarrow k) = \sum_{i=1}^{m} p_r\left(N_i^k\right) \cdot p_t\left(A \rightarrow N_i^k\right) \tag{10}$$

where N_i^k denotes the ith downstream node of A with LIC k. Functions $p_r()$ and $p_t()$ are calculated based on (6) and (7), respectively.

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

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

The probability that node A is scheduled in both channels 3 and 4, given that node A is scheduled to node B, is

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

Assuming that n takes values 1 and 2, based on (9), the equivalent fraction of the scheduled time slot that node A can use to transmit to node B is

$$\begin{aligned} p_{AB} &= 1 - \frac{1}{1+1} p_1 - \frac{2}{2+1} p_2 \\ &= 1 - \frac{1}{2} * \frac{1}{2} - \frac{2}{3} * \frac{1}{6} = \frac{23}{36}. \end{aligned} \tag{12}$$

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

Based on the aforementioned example, we can see that a node will waste no time slots 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

$$f_c = \frac{1}{p_{AB}} \tag{13}$$

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

Link cost calculation: By combining all the aforementioned major delay factors, the link cost for AB is defined as

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

Based on the aforementioned cost analysis, 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. Equation (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 a rate of W and needs $1/(1-p)$ transmissions for each packet. Therefore, the total link delay will be $O(1/(W \cdot (1-p) \cdot P_r(B) \cdot p_t(A \rightarrow B) \cdot p_{AB}))$. Because $f_r = 1/1-p$ 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 broadcasts a 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 that share 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 a *LIC* assigned when setting up other paths, it simply calculates the new accumulative subpath cost based on its *LIC*. Note that we do not assume that a centralized scheme exists to assign the channels for all the paths at the same time. Channels assigned during the previous path setup will not be modified during the new path setup. A 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 LIC, it needs to calculate the minimum cost for the subpath by inspecting every possible channel assignment for its *LI* and notes the channel that provides the minimum cost as a candidate *LIC*. The node then broadcasts 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 (14) after determining the following four major factors.

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*. Because our scheduling algorithm constrains the load of a channel in a time slot, f_r is very small and is, 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; therefore, the change should be taken into account in advance. If A is not yet an upstream node of node B, 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 the to-be-assigned channel ($channel$) for node n carried in the $nodelist$ and the possible channel assignment (ch) for B has changed. Once the information for both is obtained, node B can calculate f_s based on (8).

Algorithm 3: $AdjustedContendingNum(nodelist, ch)$

for all node $n \in nodelist$ **do**
  if (n.NodeID $\in N_B^{2-hop}$) **then**
    if (n does not 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's *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 (9).

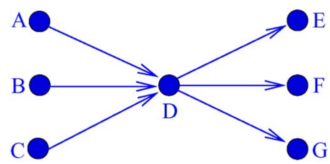


Fig. 5. Example of channel assignment and path setup.

A receiving node will not immediately tune its *LI* to the assigned channel but will wait until the path is confirmed by the destination. When the destination receives a RREQ packet, it can immediately respond with a RREP 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 that 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*.

One example is shown in Fig. 5 to explain how our channel assignment and path setup work. Assume that the data rate for each link is the same; therefore, the interface capacity factor (f_t) is constant and the same for all links. For convenience of presentation, we assume that $f_t f_r$ equals 1 and that there are two channels in the network. Initially, no node is assigned an LIC. First, source node A broadcasts 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 channels 1 and 2, respectively. Because other nodes have yet to be been assigned to a channel, the link cost is 1 for both channels 1 and 2, and thus, node D can pick either channel as the *to-be-assigned channel* (before it is confirmed by the destination). Here, we assume that channel 1 is selected, as indicated in Table I. Then, D rebroadcasts the RREQ packet, and node G receives it. Knowing from RREQ that 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. Therefore, node G will choose channel 2, and the total cost for path ADG is $1 + 1 = 2$. Because this path cost is the minimum, path ADG will be selected, and nodes D and G will be assigned channels 1 and 2, respectively. We then look at the path that searches for source node B and destination node F. Because node D is already assigned a channel during the path setup for ADG, it will keep the assignment. Assuming 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 channels 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. Note 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 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 periodically updated with the channel assignment of all its two-hop neighbors. We consider three channel-switching scenarios. The first scenario is *balancing load among channels*. If a node finds that it has many queued data for a receiver, it can notify the receiver to switch to a channel with fewer sharing neighbors. To ensure that the channel change will not increase the delay of the overloaded path, the receiver will check the cost of the path segment that passes through itself and within its two-hop range. Supposing that node C on a path $(\mathrm{A} \rightarrow \mathrm{B} \rightarrow \mathrm{C} \rightarrow \mathrm{D} \rightarrow \mathrm{E} \rightarrow \mathrm{F} \rightarrow \mathrm{G})$ finds that 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. The second scenario is *improving the 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. The third scenario is *avoiding the channel with a high error rate*. Because our scheduling algorithm constrains the number of nodes that compete 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 the channel to balance the channel and the interface usage in a neighborhood also helps improve fairness among neighboring nodes.

2) Replace Operation: If a node has either a *TI* or *LI bottleneck*, it will look for an alternative path that goes through a *replacement node* to forward the data. The replacement node should ensure that the new path that passes through itself will not have a higher end-to-end delay than the old path. Given a path segment $(\mathrm{A} \rightarrow \mathrm{B} \rightarrow \mathrm{C} \rightarrow \mathrm{D} \rightarrow \mathrm{E})$, if C has an interface bottleneck, C will check the path that passes through a neighboring node within B and D's transmission range, e.g., a node F. Node C will compare the total cost for $(\mathrm{A} \rightarrow \mathrm{B} \rightarrow \mathrm{F} \rightarrow \mathrm{D} \rightarrow \mathrm{E})$ with the cost of the current path segment. If the new cost is smaller, node C will send the message to nodes 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 $(\mathrm{A} \rightarrow \mathrm{B} \rightarrow \mathrm{C})$, if node A detects that both B and C are its one-hop neighbors, it can directly forward the data packets to node C.

4) Insert Operation: Given a path segment $(\mathrm{A} \rightarrow \mathrm{B})$, 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 with good quality, it can insert itself between nodes A and B.

To reduce the implementation cost, the aforementioned 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 a mobility scenario.

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**	

VI. PERFORMANCE EVALUATION

We implemented our proposed algorithms using the simulation package GloMoSim [27]. Each node is assumed to have only two IEEE 802.11a interfaces, with an interface rate of 54 Mb/s. The time slot length is set to 10 ms (about 35 maximum-length packet transmission time [11]), the broadcast interval of hello messages is set to 5 s, and the backoff window sizes for $W0$, $W1$, and $W2$ in the prioritized transmitting scheme (see Section IV) are set to 7, 15, and 31, respectively. The transmission power is 15 dBm, the radio sensitivity is -84 dBm, and the radio receiving threshold is -74 dBm. We compare the performance using our integrated MAC and routing framework with the scheme that uses independent MAC and routing, e.g., dynamic channel assignment (DCA) [9] as MAC and AODV as routing, as well as the scheme that simply uses AODV over IEEE 802.11a. One reason for selecting DCA is because it also uses two interfaces, which can provide a fairer comparison, compared with schemes that use only a single interface or the schemes that use the number of interfaces larger than two. In the DCA scheme, one of the channels is used as the control channel, whereas 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 the control channel to select a channel to transmit data. Then, the sender
AQ4 broadcasts a resume (RES) message over the control channel to reserve the data channel and sends the data packet to the receiver.

AQ5 Constant bit rate (CBR) is used as the application protocol. To provide enough traffic load to study the multichannel benefit, the size of a packet is set as 2000 B, and packets are sent out every 0.5 ms. Each simulation runs 100 s. For each run, we try to get the maximum throughput by tuning CBR 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 two, three, four, and five 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 the DCA scheme, and the AODV over the 802.11a scheme, respectively.

A. Chain-topology

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

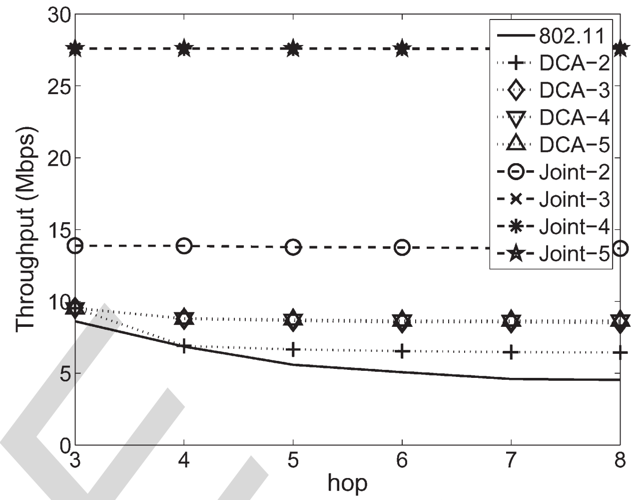


Fig. 6. Throughput in the chain topology.

If there are only two channels, similar to 802.11, DCA can only use one channel for data transmission. However, by separating the 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 the control channel, and the remaining two channels will be used as data channels. In a snapshot of the network, the best channel assignment for the links along the chain could be, e.g., "…, channel 1, idle, channel 2, idle, channel 1, idle, … ." The link between two active links is kept idle, because 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 the throughput. Thus, the curves of DCA-3, DCA-4, and DCA-5 overlap in Fig. 6.

In contrast, our protocol can make better use of more channels. If there are only two channels, in a network snapshot, the best channel usage for the links along the chain could be, e.g., "…, channel 1, channel 2, idle, channel 1, channel 2, idle, … ." With three channels, our protocol could achieve better throughput. The network snapshot could be, e.g., "…, 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 5×5 grid network. The

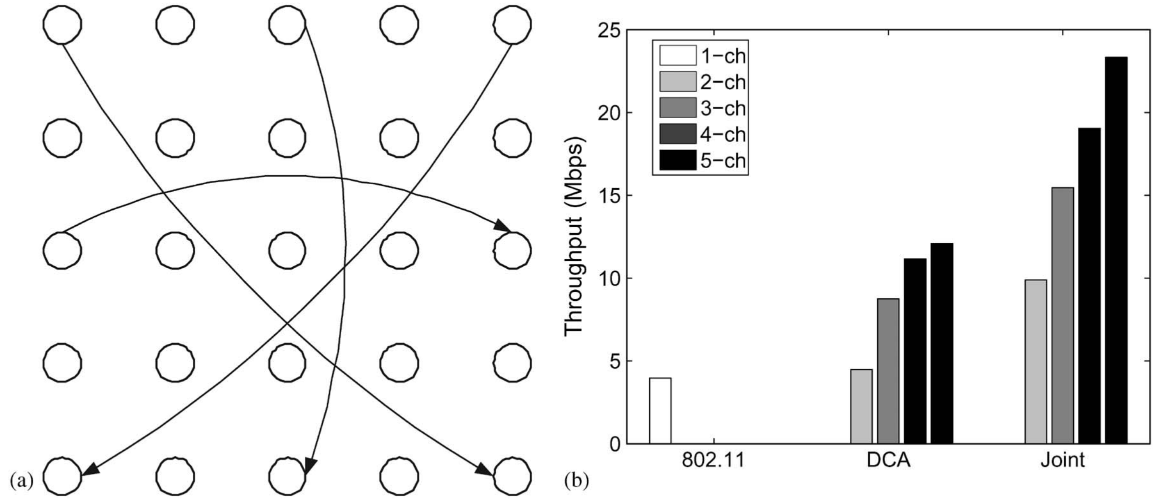


Fig. 7. Performance for the grid topology. (a) Topology. (b) Throughput.

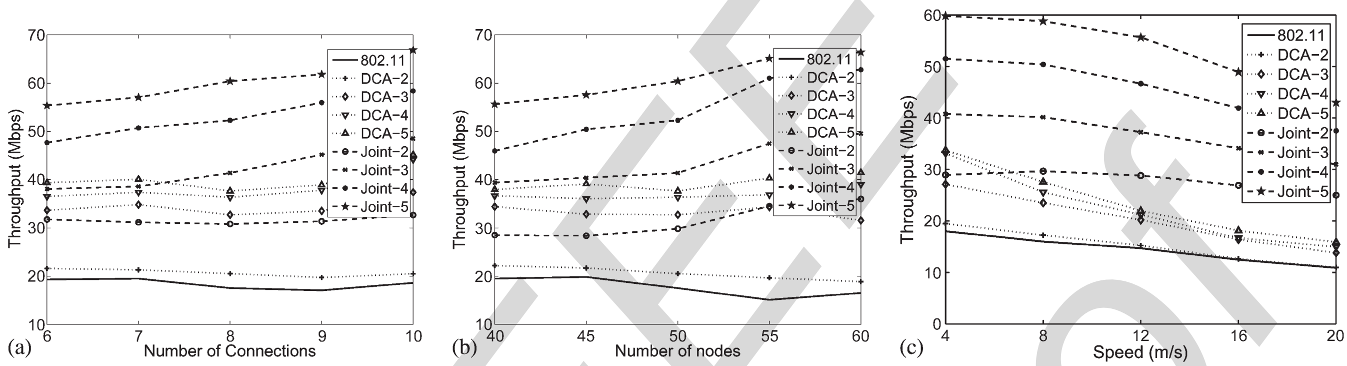


Fig. 8. Throughput for the random topology. (a) Effect of load. (b) Effect of node density. (c) Effect of mobility.

grid distance is set such that the receiving power at a neighboring node is −70 dBm. 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 the aggregate network throughput are shown in Fig. 7(b).

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

C. Random Topology

In this set of simulations, nodes can randomly move within a 1000 × 1000 m network area. The movement follows the improved random waypoint model [28]. Because we use 802.11a, which has a lower transmission range than 802.11 b, the default average moving speed is set to 5 m/s, and the maximum speed is set to 10 m/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 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 numbers of channels are much higher than those using other schemes. The aggregate throughputs for both 802.11 and DCA-2 (with one data channel) decrease as the number of connections increases. This result 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, because 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 slightly increases, because the network is saturated with only two channels. With more channels, the throughput of our protocol has a larger increase at a higher load compared with DCA, because our protocol can more efficiently handle additional connections 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 a much higher throughput increase as the node density increases, whereas 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 the results 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 paths and route traffic away from bottlenecks during route maintenance.

Finally, 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 4 m/s to 20 m/s. The simulation results for aggregate throughput are shown in Fig. 8(c). As expected, the throughput for all three protocols decreases when the speed increases as a result of the link breakage during mobility. In addition, the decrease is faster when more channels are used. Because 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 more slowly than the reference schemes, which indicate that our maintenance scheme can effectively adapt the path and channel assignment to topology changes, thus preventing link breakage in advance.

VII. Conclusion

In this paper, we have proposed an integrated MAC and routing design to explore the capabilities provided by multiple channels and multiple interfaces in ad hoc networks. We defined 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 a limited number of channels. Based on the routing metric, we proposed a routing algorithm for path discovery, which considers all the major factors of a MCMI network in finding the minimum cost path. We also presented route maintenance schemes for adapting 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 the 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 minimize the effect of channel switching delay, balance the load, and enable fairness among neighboring nodes. In addition, we enhanced 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 a 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 very efficiently utilize the channel resources to significantly improve the network throughput in a multichannel multi-interface environment.

References

[1] A. Raniwala, K. Gopalan, and T. Chiueh, "Centralized algorithms for multichannel wireless mesh networks," *ACM Mobile Comput. Commun. Rev.*, vol. 8, no. 2, pp. 50–65, Apr. 2004.

[2] A. Raniwala and T. Chiueh, "Algorithms for an IEEE-802.11-based multichannel wireless mesh network," in *Proc. IEEE INFOCOM*, Mar. 2005, pp. 2223–2234.

[3] M. Alicherry, R. Bhatia, and L. Li, "Joint channel assignment and routing for throughput optimization in multiradio wireless mesh networks," in *Proc. ACM MobiCom*, Sep. 2005, pp. 58–72.

[4] A. Mishra, D. Agrawal, V. Shrivastava, S. Banerjee, and S. Ganguly, "Distributed channel management in uncoordinated wireless environments," in *Proc. ACM MobiCom*, Los Angeles, CA, Sep. 2006, pp. 170–181.

[5] K. N. Ramachandran, E. M. Belding, K. C. Almeroth, and M. M. Buddhikot, "Interference-aware channel assignment in multiradio wireless mesh networks," in *Proc. IEEE INFOCOM*, Apr. 2006, pp. 1–12.

[6] M. Kodialam and T. Nandagopal, "Characterizing the capacity region in multiradio, multichannel wireless mesh networks," in *Proc. ACM MobiCom*, Sep. 2005, pp. 73–87.

[7] A. Nasipuri and S. R. Das, "Multichannel CSMA with signal-power-based channel selection for multihop wireless networks," in *Proc. IEEE VTC*, Sep. 2000, pp. 211–218.

[8] N. Jain and S. R. Das, "Protocol with receiver-based channel selection for multihop wireless networks," in *Proc. IEEE IC^{3}N*, Oct. 2001, pp. 432–439.

[9] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, and J.-P. Sheu, "New multichannel MAC protocol with on-demand channel assignment for mobile ad hoc networks," in *Proc. I-SPAN*, Oct. 2000, pp. 232–237.

[10] J. So and N. H. Vaidya, "Multichannel MAC for ad hoc networks: Handling multichannel hidden terminals using a single transceiver," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, May 2004, pp. 222–233.

[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 *Proc. ACM MobiCom*, Sep. 2004, pp. 216–230.

[12] A. Tzamaloukas and J. J. Garcia-Luna-Aceves, "A receiver-initiated collision-avoidance protocol for multichannel networks," in *Proc. IEEE INFOCOM*, Apr. 2001, pp. 189–198.

[13] J. So and N. H. Vaidya, "A routing protocol for utilizing multiple channels in multihop wireless networks with a single transceiver," Univ. Illinois Urbana-Champaign, Champaign, IL, Tech. Rep., Oct. 2004.

[14] P. Kyasanur and N. Vaidya, "Routing and link-layer protocols for multichannel multi-interface ad hoc wireless networks," *SIGMOBILE Mobile Comput. Commun. Rev.*, 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 multichannel ad hoc wireless networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 6, pp. 1874–1887, Dec. 2009.

[16] S. Merlin, N. H. Vaidya, and M. Zorzi, "Resource allocation in multiradio multichannel multihop wireless networks," in *Proc. IEEE INFOCOM*, Apr. 2008, pp. 610–618.

[17] J. Shi, T. Salonidis, and E. W. Knightly, "Starvation mitigation through multichannel coordination in CSMA multihop wireless networks," in *Proc. 7th ACM Int. Symp. Mobile Ad Hoc Netw. Comput., MobiHoc*, 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)," IETF MANET Working Group (Draft 10), 2004.

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

[20] R. Draves, J. Padhye, and B. Zill, "Routing in multiradio, multihop wireless mesh networks," in *Proc. ACM MobiCom*, Sep. 2004, pp. 114–128.

[21] W.-H. Tarn and Y. C. Tseng, "Joint multichannel link layer and multipath routing design for wireless mesh networks," in *Proc. IEEE INFOCOM*, May 2007, pp. 2081–2089.

[22] D. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multihop wireless routing," in *Proc. ACM MobiCom*, Sep. 2003, pp. 134–146.

[23] H. Li, Y. Cheng, C. Zhou, and W. Zhuang, "Minimizing end-to-end delay: A novel routing metric for multiradio wireless mesh networks," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 46–54.

[24] L. Bao and J. Garcia-Luna-Aceves, "Hybrid channel access scheduling in ad hoc networks," in *Proc. IEEE ICNP*, Nov. 2002, pp. 46–57.

[25] J. Padhye, S. Agarwal, V. N. Padmanabhan, L. Qiu, A. Rao, and B. Zill, "Estimation of link interference in static multihop wireless networks," in *Proc. IMC*, Oct. 2005, p. 28.

[26] G. Bianchi and I. Tinnirello, "Kalman filter estimation of the number of competing terminals in an IEEE 802.11 network," in *Proc. IEEE INFOCOM*, Mar. 2003, pp. 844–852.
[27] X. Zeng, R. Bagrodia, and M. Gerla, "GLOMOSIM: A library for parallel simulation of large-scale wireless networks," in *Proc. 12th Workshop PADS*, May 1998, pp. 154–161.
[28] W. Navidi and T. Camp, "Stationary distributions for the random waypoint mobility model," *IEEE Trans. Mobile Comput.*, vol. 3, no. 1, pp. 99–108, Jan./Feb. 2004.

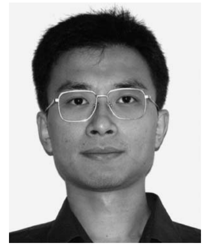
Jinhua Zhu received the Ph.D. degree in computer science and engineering from the University at Buffalo, Buffalo, NY, in 2005.

He is currently a Service Engineer with Ericsson Inc., Boulder, CO. His research interests include wireless sensor networks and cognitive radio.

Xin Wang (M'01) received the Ph.D. degree in electrical and computer engineering from Columbia University, New York, NY, in 2001.

From 2001 to 2003, she was a member of Technical Staff with Bell Labs Research, Lucent Technologies, Murray Hill, NJ, and an Assistant Professor with the Department of Computer Science and Engineering, University at Buffalo, Buffalo, NY. She is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY. Her research interests include wireless networks and communications and mobile and distributed computing.

Dr. Wang has served in numerous technical communities of top networking conferences.



Dahai Xu (S'01–M'05) received the Ph.D. degree from the University at Buffalo, Buffalo, NY, in 2005.

He spent two years as a Postdoctoral Research Associate with Princeton University, Princeton, NJ. He is currently a Research Staff with AT&T Labs Research, Florham Park, NJ. His research interests include Internet design, control and management, algorithm design and fast implementation, large-scale nonlinear network optimization, and secure communication in wireless ad hoc networks.

AUTHOR QUERIES

AUTHOR PLEASE ANSWER ALL QUERIES

AQ1 = T/R was defined as transmit-receive. Please check if this is correct. Otherwise, provide the corresponding definition.

AQ2 = WCETT was defined as weighted cumulative expected transmission time. Please check if this is correct. Otherwise, provide the corresponding definition.

AQ3 = RREQ, RRER, and RREP were defined as route request, route error, and route reply, respectively. Please check if these are correct. Otherwise, provide the corresponding definitions.

AQ4 = RES was defined as resume. Please check if this is correct. Otherwise, provide the corresponding definition.

AQ5 = CBR was defined as constant bit rate. Please check if this is correct. Otherwise, provide the corresponding definition.

END OF ALL QUERIES

A Unified MAC and Routing Framework for Multichannel Multi-interface Ad Hoc Networks

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

***Abstract*—Improving the capacity of wireless networks is critical and challenging. Although wireless standards such as IEEE 802.11 allow the use of multiple channels at the physical layer, current Media Access Control (MAC) and routing protocols of mobile ad hoc networks have mainly been developed to run 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, as well as feedback from the MAC layer. Given the routing path and channel assignment, our scheduling scheme at the MAC layer explores the resources at the time domain to coordinate transmissions within an interference range to maximize channel usage, reduce channel access competition among nodes assigned to 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 to reduce the transmission delay of mission-critical packets and message broadcast, which help 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 multichannel multi-interface ad hoc networks.**

***Index Terms*—Ad hoc networks, cross layer, Media Access Control (MAC), multichannel, multiradio, routing.**

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 local area network (LAN) standards such as IEEE 802.11 often allow for transmissions on multiple physical channels, current Media Access Control (MAC) and routing protocols in infrastructure-free ad hoc networks are generally designed to transmit data only on one channel. In addition, 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 that 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 paper 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 multichannel multi-interface (MCMI) environment. Because 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 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. Aside from these issues, in a multihop network, it is critical and challenging to establish a routing path that exploits the MCMI feature for better throughput and to 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 multichannel environment.

Because the aforementioned 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, i.e., to exploit MCMI capabilities in mobile ad hoc networks to fully use the available spectrum to improve the network performance. Our framework jointly considers routing and 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

Manuscript received October 28, 2009; revised January 11, 2010 and June 12, 2010; accepted September 5, 2010. This work was supported in part by the U.S. National Science Foundation under Grant CNS 0751121 and Grant CNS 0628093. The review of this paper was coordinated by Prof. A. Boukerche.

J. Zhu is with Ericsson Inc., Boulder, CO 80301 USA.

X. Wang is with the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794 USA.

D. Xu is with the AT&T Labs Research, Florham Park, NJ 07932 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2010.2080362

the path. Our route-maintenance scheme adopts the path and channel assignment based on changes of topology and channel condition and on feedbacks from the MAC layer. Given the channel assignments during path setup, a *scheduling scheme* is used at the MAC layer to coordinate the channel usage and interface sharing/switching to enable communications between nodes and 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 interfaces is smaller than the number of available channels. Our contributions can be summarized as follows:

- Design 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.
- Develop 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.
- Incorporate 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.
- Design a scheduling scheme that manages resources in the time dimension to coordinate channel usage and interface sharing among neighboring nodes assigned the same channel to reduce channel competitions, to 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.
- Enhance 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 multichannel broadcast delay and the transmission delay for mission critical applications, and allow unscheduled nodes to opportunistically use the available channel resources to improve throughput.

Multichannel multiradio wireless networks have received a substantial amount of recent interest, particularly 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]. Because 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, whereas using single interface [13] in multichannel environment for routing would result in poor connectivity.

To the best of our knowledge, this paper provides the first practical network framework that concurrently considers routing and channel assignment at the 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 to avoid the deafness problem when a transmission pair tunes their radio interfaces to the same channel at different times. 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 the 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 the delay of mission-critical traffic and control messages.

The rest of this 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 that pertain to a MCMI 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 detail a joint routing and channel assignment scheme and an efficient channel and route-maintenance scheme. Section VI describes our evaluation using simulations. We conclude this 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, whereas 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] and [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] and [11], the authors propose multiple access schemes for the nodes equipped with single interface. Receiver-initiated channel-hopping with dual polling (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, whereas the scheme in [17] targets addressing the starvation problem in a Carrier Sense Multiple Access (CSMA)-based multihop wireless network.

Predominant routing protocols such as dynamic source routing (DSR) [18] and ad hoc on-demand distance vector (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, because 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 interdependent and that transmission on the same path may experience intrachannel 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 apply in the mobile ad hoc networks, which require a distributed scheme to quickly react to topology change. The scheme proposed in [21] combines multichannel link layer with multipath routing. Although interesting, many design ideas [e.g., superframe pattern, dynamic adjustment of the transmit–receive (T/R) ratio, and multipath routing] proposed in this paper target to address the inefficiency due to the half-duplex transmissions as a result of using one radio interface at each node. The use of a single interface would lead to more severe multichannel hidden terminal problem [10] and deafness problem. In [20], the authors extend the work in [22] and propose a new routing metric, i.e., weighted cumulative expected transmission time (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 intrapath 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 that the channel has been assigned, the work in [23] considers queuing 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 the 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 for implementing the algorithms. Although the proposed scheme is not centralized, a supernode 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 neighborhoods 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 an MCMI network. The scheduling algorithm is purely distributed, and each node can make a scheduling decision to efficiently coordinate channel usage and interface switching with no need for complicated signaling messages.

III. System Overview

The goal of this paper 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 the *frequency* domain through channel assignment and the *time* domain through transmission time slot scheduling to significantly increase the network throughput. Our design at the routing layer includes the following techniques: 1) a *link cost model* for capturing the characteristics of MCMI networks and the impact of MAC-layer scheduling; 2) a *joint channel assignment and routing scheme* for concurrently searching for the minimum cost path and assigning channels to nodes along the path; and 3) a route-maintenance scheme for adapting the path and channel assignment in response to changes of network topology and channel conditions and MAC feedback. Given channels assigned during the path setup, our design at the MAC layer includes the following techniques: 1) a *distributed scheduling scheme* for coordinating the channel usage in the unit of time slot to reduce competition among nodes assigned the same channel within an interference range and for coordinating interface sharing and switching to reduce transmission conflict and unnecessary switching delay and 2) a *prioritized transmission scheme* for coordinating 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 multichannel 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 the following two types: 1) the *listening interface* (*LI*) and 2) the *transmitting interface* (*TI*). During path setup, one radio interface of a node will be designated as *LI* and assigned a channel, called the *LI channel* (*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 TIs, which can flexibly be 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 periodically be sent by a node to maintain network topology, as is 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 quickly reduces as the distance between the transmitter and receiver increases. To reduce the implementation overhead, in this paper, we consider interference of 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*.

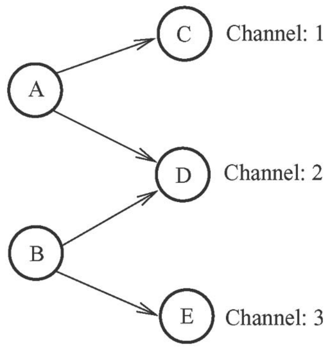


Fig. 1. Example of transmission coordination.

IV. Media Access Control 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 the delay of important control and data packets. Our MAC scheme addresses the following issues.

1) *Interference among the transmissions over the same channel*. There are generally a limited number of channels in the system. Due to cost, time, and policy constraints, the number of channels to which a node can tune 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 the use of multiple channels, an interface needs to be switched among different channels. Because channel switching incurs a nonignorable delay [11], it would be more efficient to reduce channel switching.
3) *Transmission conflict*. A node may have several downstream nodes that listen to different channels. With no coordination, independent transmissions from multiple upstream nodes to the same channel will result in collisions, whereas better channel usage coordination would lead to concurrent transmissions. For example, in Fig. 1, node A can transmit to nodes C and D using channels 1 and 2, respectively, whereas node B can transmit to nodes D and E using channels 2 and 3, respectively. Without any coordination, nodes A and B may try to transmit to node D using channel 2 at the same time, whereas neither channel 1 nor channel 3 is used, which causes both collision at the same receiver and channel resource wastage.
4) *Broadcast delay*. Because 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 condition 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).

A. Channel-Scheduling Scheme

In a MCMI system, a simple exchange of request to send/clear to send (RTS/CTS) between a sender and a receiver on the LIC of the receiver is not enough to avoid the hidden terminal problem, because a potential interference node may be listening to a different channel, whereas 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 that 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 LIC, 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; and 3) when a node is scheduled to transmit to multiple receivers with different LICs, 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-A2, 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 (see 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 significant throughput degradation under heavy load as in a pure CSMA with collision avoidance (CSMA/CA)-based scheme such as IEEE 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. Because 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 of 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. The data packet from the sender to the receiver is generally longer than the confirmation packet from the receiver to the sender; therefore, it is more important to reduce interference at the receiver side. Our scheduling has the following two phases: 1) *receiver scheduling* and 2) *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 the 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 *with no need for 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 has the highest priority, the entity with the largest ID will be selected.

This algorithm is summarized in 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 that contend for the same *LIC* (ch) as i. $Rand(X, t)$ is adopted from the $Hash()$ function used in [24]. We have

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

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 that the size of the output of $Hash()$ function is 64 b. 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.

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

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

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 of its two-hop neighbors that 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 that 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 a nonoverlapping *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.

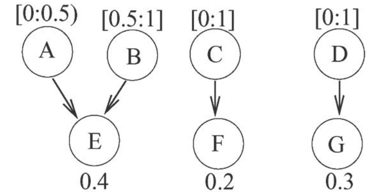


Fig. 2. Example of scheduling.

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

if ($Rand(i, t) \in P_{i,j}$) **then**
    return TRUE
else
    return FALSE
end if

One example is shown in Fig. 2 to explain how our scheduling works. There are four senders (nodes A, B, C, and D) and three receivers (nodes 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 (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. Because 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. Because node E's random value (0.4) falls within node A's probability range, i.e., [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 node B is not scheduled to transmit to node E. Nodes C and D determine that nodes 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

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

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

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

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 a 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 frequently happen, because 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 the 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 is 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, which will be discussed as follows.

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-A2). With the support of time-slot-based scheduling, the following additional issues should still be addressed.

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., route request (RREQ), route error (RRER), and route reply (RREP) packets] and mission-critical data packets. To avoid a collision in transmitting the first priority packets, each node waits for some random time within a window $W0$.

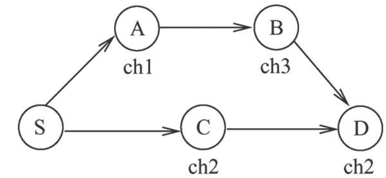


Fig. 3. Two possible paths.

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 an RTS packet but with a different delay window $W1$ larger than $W0$.

The *third* level of priority will be assigned to the nonscheduled senders to avoid wasting the time slots that cannot be used up by the scheduled transmissions. To avoid competing with the scheduled sender, a nonscheduled 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 nonscheduled nodes only need to wait for a random period of time within the window $W2$ before transmitting subsequent packets. In addition, a nonscheduled 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.

Note that our scheme is robust in the 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 the 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 nonscheduled nodes with a lower priority. We will show in the next section that more than one node within a 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 nodes C and D in Fig. 3. Assume that each link has the same transmission rate. Although path SCD has only two hops, because nodes C and D are assigned the same LIC ($ch2$), 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 nodes 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 paper, 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.

A routing protocol generally searches for the minimum cost path between the source and the destination. Because 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 nonlinear 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 suboptimal 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 the 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., 11Mb/s in IEEE 802.11b and 54Mb/s 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, because 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. To measure the condition of a channel, there is also a need to first transmit data on the channel, which may not be possible before the channel is assigned. The interference measurement in [25] can be only used for static networks. Instead, we analytically estimate the packet error rate based on our scheduling scheme.

Assume that the interference range is about twice the transmission range. In our scheduling scheme, only one receiver is scheduled within a two-hop neighborhood. Assuming that the network area is A, the transmission range is R, and the nodes are evenly distributed. If the scheduled receivers are at 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

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

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 IEEE 802.11, as described in Section IV-B. Most transmission failures are due to collisions (e. g., collisions in RTS messages). For an IEEE 802.11 network, the collision probability or packet error rate p is impacted by the number of contending nodes n [26], i.e.,

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

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

Because our scheduling algorithm restricts the average number of competing nodes within the interference range to be a constant number 4, based on (4), the average packet error rate p is small and a constant. The expected number of transmissions (*ETX*) can be calculated as $1/(1-p)$. 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) as follows:

$$f_r = \frac{1}{1-p}. \tag{5}$$

Because 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. Although scheduling helps 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 that share the same *LIC* within a two-hop neighborhood, only one node is scheduled for receiving in a slot. Assuming that each node has the same probability of being scheduled for receiving and node B is assigned channel ch as its *LIC*, the probability that node B is scheduled for receiving in channel ch is

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

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

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

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

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

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

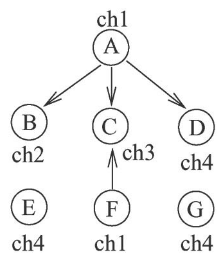


Fig. 4. Transmission-conflicting example.

This factor reflects the impact of network topology and channel constraint on the network throughput. If there are a large number of nodes that share 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 the 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, because scheduling is distributedly performed in reference to each receiver, it may be scheduled to transmit 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 channels 2, 3, and 4, respectively. Node A is also scheduled to transmit to all the three nodes. Because 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}, i.e., 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.

The concept of equivalent fraction of the time slot can be understood in an intuitive way. We assume that 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. Suppose that 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, $\sum_n (p_n n/n+1)$ part of the time slot will be used to transmit in the channels other than channel ch. One example is given in the latter part of this section to show how we can calculate the probability p_n. The fraction of the time slot that node A can use to transmit to node B in channel ch can be calculated as

$$p_{AB} = 1 - \sum_n p_n \frac{n}{n+1}. \tag{9}$$

To calculate the equivalent fraction, we consider two cases:

Case 1: 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, because two interfaces of a node cannot be tuned to the same channel for transmitting and receiving at the same time. Because both nodes' *LIs* 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 can always use its *LI* to transmit, regardless of the channel usage of node A's *TI*. That is, node A can use all portions of the scheduled time slot, i.e., $p_{AB} = 1$.

Case 2: 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 also scheduled to transmit to other nodes (we call it *conflicting probability*).

To calculate p_{AB} based on (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*. Assuming that node A has m downstream nodes, which are assigned the same LIC k, the probability that node A is scheduled to transmit on channel k is

$$p_{tch}(A \Rightarrow k) = \sum_{i=1}^{m} p_r\left(N_i^k\right) \cdot p_t\left(A \rightarrow N_i^k\right) \tag{10}$$

where N_i^k denotes the ith downstream node of A with LIC k. Functions $p_r()$ and $p_t()$ are calculated based on (6) and (7), respectively.

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

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

The probability that node A is scheduled in both channels 3 and 4, given that node A is scheduled to node B, is

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

Assuming that n takes values 1 and 2, based on (9), the equivalent fraction of the scheduled time slot that node A can use to transmit to node B is

$$\begin{aligned} p_{AB} &= 1 - \frac{1}{1+1} p_1 - \frac{2}{2+1} p_2 \\ &= 1 - \frac{1}{2} * \frac{1}{2} - \frac{2}{3} * \frac{1}{6} = \frac{23}{36}. \end{aligned} \tag{12}$$

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

Based on the aforementioned example, we can see that a node will waste no time slots 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

$$f_c = \frac{1}{p_{AB}} \tag{13}$$

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

Link cost calculation: By combining all the aforementioned major delay factors, the link cost for AB is defined as

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

Based on the aforementioned cost analysis, 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. Equation (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 $p_t(A \to B)$ 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 a rate of W and needs $1/(1-p)$ transmissions for each packet. Therefore, the total link delay will be $O(1/(W \cdot (1-p) \cdot P_r(B) \cdot p_t(A \to B) \cdot p_{AB}))$. Because $f_r = 1/1-p$ 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 broadcasts a 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 that share 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 a *LIC* assigned when setting up other paths, it simply calculates the new accumulative subpath cost based on its *LIC*. Note that we do not assume that a centralized scheme exists to assign the channels for all the paths at the same time. Channels assigned during the previous path setup will not be modified during the new path setup. A 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 LIC, it needs to calculate the minimum cost for the subpath by inspecting every possible channel assignment for its *LI* and notes the channel that provides the minimum cost as a candidate *LIC*. The node then broadcasts 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 (14) after determining the following four major factors.

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*. Because our scheduling algorithm constrains the load of a channel in a time slot, f_r is very small and is, 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; therefore, the change should be taken into account in advance. If A is not yet an upstream node of node B, 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 the to-be-assigned channel ($channel$) for node n carried in the $nodelist$ and the possible channel assignment (ch) for B has changed. Once the information for both is obtained, node B can calculate f_s based on (8).

Algorithm 3: $AdjustedContendingNum(nodelist, ch)$
for all node $n \in nodelist$ **do**
    if (n.NodeID $\in N_B^{2-hop}$) **then**
        if (n does not 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's *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 (9).

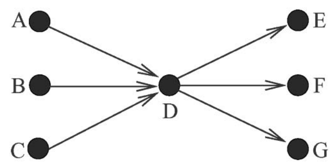


Fig. 5. Example of channel assignment and path setup.

A receiving node will not immediately tune its *LI* to the assigned channel but will wait until the path is confirmed by the destination. When the destination receives a RREQ packet, it can immediately respond with a RREP 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 that 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*.

One example is shown in Fig. 5 to explain how our channel assignment and path setup work. Assume that the data rate for each link is the same; therefore, the interface capacity factor (f_t) is constant and the same for all links. For convenience of presentation, we assume that $f_t f_r$ equals 1 and that there are two channels in the network. Initially, no node is assigned an LIC. First, source node A broadcasts 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 channels 1 and 2, respectively. Because other nodes have yet to be been assigned to a channel, the link cost is 1 for both channels 1 and 2, and thus, node D can pick either channel as the *to-be-assigned channel* (before it is confirmed by the destination). Here, we assume that channel 1 is selected, as indicated in Table I. Then, D rebroadcasts the RREQ packet, and node G receives it. Knowing from RREQ that 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. Therefore, node G will choose channel 2, and the total cost for path ADG is $1 + 1 = 2$. Because this path cost is the minimum, path ADG will be selected, and nodes D and G will be assigned channels 1 and 2, respectively. We then look at the path that searches for source node B and destination node F. Because node D is already assigned a channel during the path setup for ADG, it will keep the assignment. Assuming 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 channels 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. Note 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 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 periodically updated with the channel assignment of all its two-hop neighbors. We consider three channel-switching scenarios. The first scenario is *balancing load among channels*. If a node finds that it has many queued data for a receiver, it can notify the receiver to switch to a channel with fewer sharing neighbors. To ensure that the channel change will not increase the delay of the overloaded path, the receiver will check the cost of the path segment that passes through itself and within its two-hop range. Supposing that node C on a path $(\mathrm{A} \rightarrow \mathrm{B} \rightarrow \mathrm{C} \rightarrow \mathrm{D} \rightarrow \mathrm{E} \rightarrow \mathrm{F} \rightarrow \mathrm{G})$ finds that 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. The second scenario is *improving the 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. The third scenario is *avoiding the channel with a high error rate*. Because our scheduling algorithm constrains the number of nodes that compete 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 the channel to balance the channel and the interface usage in a neighborhood also helps improve fairness among neighboring nodes.

2) Replace Operation: If a node has either a *TI* or *LI bottleneck*, it will look for an alternative path that goes through a *replacement node* to forward the data. The replacement node should ensure that the new path that passes through itself will not have a higher end-to-end delay than the old path. Given a path segment $(\mathrm{A} \rightarrow \mathrm{B} \rightarrow \mathrm{C} \rightarrow \mathrm{D} \rightarrow \mathrm{E})$, if C has an interface bottleneck, C will check the path that passes through a neighboring node within B and D's transmission range, e.g., a node F. Node C will compare the total cost for $(\mathrm{A} \rightarrow \mathrm{B} \rightarrow \mathrm{F} \rightarrow \mathrm{D} \rightarrow \mathrm{E})$ with the cost of the current path segment. If the new cost is smaller, node C will send the message to nodes 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 $(\mathrm{A} \rightarrow \mathrm{B} \rightarrow \mathrm{C})$, if node A detects that both B and C are its one-hop neighbors, it can directly forward the data packets to node C.

4) Insert Operation: Given a path segment $(\mathrm{A} \rightarrow \mathrm{B})$, 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 with good quality, it can insert itself between nodes A and B.

To reduce the implementation cost, the aforementioned 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 a mobility scenario.

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**	

VI. Performance Evaluation

We implemented our proposed algorithms using the simulation package GloMoSim [27]. Each node is assumed to have only two IEEE 802.11a interfaces, with an interface rate of 54 Mb/s. The time slot length is set to 10 ms (about 35 maximum-length packet transmission time [11]), the broadcast interval of hello messages is set to 5 s, and the backoff window sizes for $W0$, $W1$, and $W2$ in the prioritized transmitting scheme (see Section IV) are set to 7, 15, and 31, respectively. The transmission power is 15 dBm, the radio sensitivity is -84 dBm, and the radio receiving threshold is -74 dBm. We compare the performance using our integrated MAC and routing framework with the scheme that uses independent MAC and routing, e.g., dynamic channel assignment (DCA) [9] as MAC and AODV as routing, as well as the scheme that simply uses AODV over IEEE 802.11a. One reason for selecting DCA is because it also uses two interfaces, which can provide a fairer comparison, compared with schemes that use only a single interface or the schemes that use the number of interfaces larger than two. In the DCA scheme, one of the channels is used as the control channel, whereas 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 the control channel to select a channel to transmit data. Then, the sender broadcasts a resume (RES) message over the control channel to reserve the data channel and sends the data packet to the receiver.

Constant bit rate (CBR) is used as the application protocol. To provide enough traffic load to study the multichannel benefit, the size of a packet is set as 2000 B, and packets are sent out every 0.5 ms. Each simulation runs 100 s. For each run, we try to get the maximum throughput by tuning CBR 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 two, three, four, and five 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 the DCA scheme, and the AODV over the 802.11a scheme, respectively.

A. Chain-topology

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

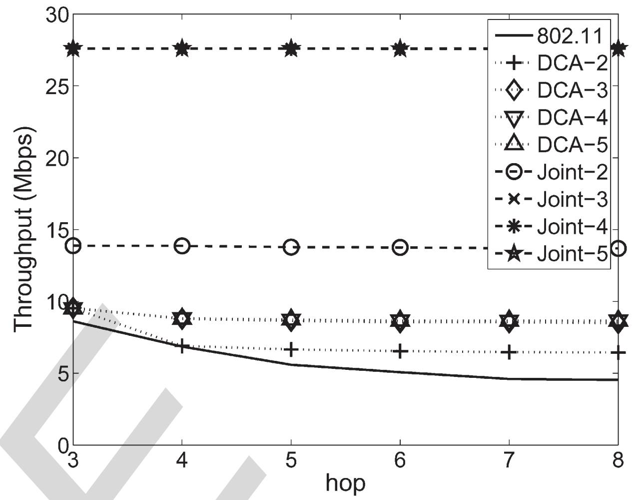


Fig. 6. Throughput in the chain topology.

If there are only two channels, similar to 802.11, DCA can only use one channel for data transmission. However, by separating the 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 the control channel, and the remaining two channels will be used as data channels. In a snapshot of the network, the best channel assignment for the links along the chain could be, e.g., "…, channel 1, idle, channel 2, idle, channel 1, idle, …." The link between two active links is kept idle, because 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 the throughput. Thus, the curves of DCA-3, DCA-4, and DCA-5 overlap in Fig. 6.

In contrast, our protocol can make better use of more channels. If there are only two channels, in a network snapshot, the best channel usage for the links along the chain could be, e.g., "…, channel 1, channel 2, idle, channel 1, channel 2, idle, …." With three channels, our protocol could achieve better throughput. The network snapshot could be, e.g., "…, 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 5×5 grid network. The

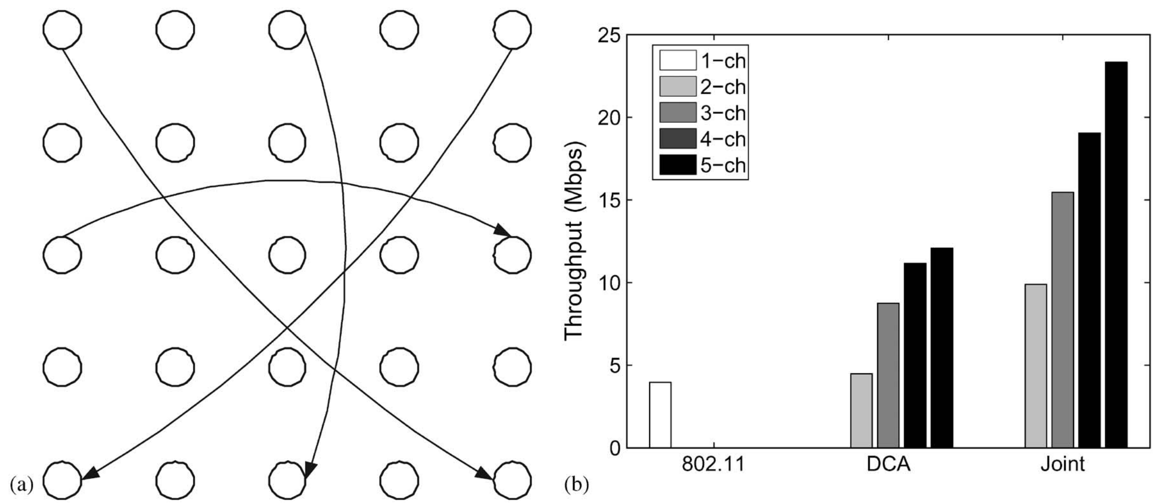


Fig. 7. Performance for the grid topology. (a) Topology. (b) Throughput.

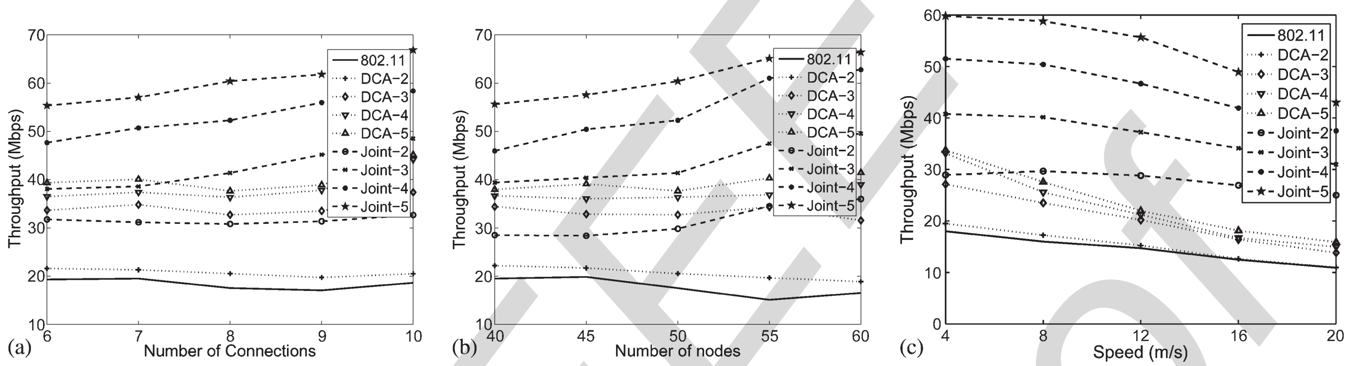


Fig. 8. Throughput for the random topology. (a) Effect of load. (b) Effect of node density. (c) Effect of mobility.

grid distance is set such that the receiving power at a neighboring node is −70 dBm. 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 the aggregate network throughput are shown in Fig. 7(b).

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

C. Random Topology

In this set of simulations, nodes can randomly move within a 1000 × 1000 m network area. The movement follows the improved random waypoint model [28]. Because we use 802.11a, which has a lower transmission range than 802.11 b, the default average moving speed is set to 5 m/s, and the maximum speed is set to 10 m/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 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 numbers of channels are much higher than those using other schemes. The aggregate throughputs for both 802.11 and DCA-2 (with one data channel) decrease as the number of connections increases. This result 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, because 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 slightly increases, because the network is saturated with only two channels. With more channels, the throughput of our protocol has a larger increase at a higher load compared with DCA, because our protocol can more efficiently handle additional connections 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 a much higher throughput increase as the node density increases, whereas 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 the results 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 paths and route traffic away from bottlenecks during route maintenance.

Finally, 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 4 m/s to 20 m/s. The simulation results for aggregate throughput are shown in Fig. 8(c). As expected, the throughput for all three protocols decreases when the speed increases as a result of the link breakage during mobility. In addition, the decrease is faster when more channels are used. Because 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 more slowly than the reference schemes, which indicate that our maintenance scheme can effectively adapt the path and channel assignment to topology changes, thus preventing link breakage in advance.

VII. Conclusion

In this paper, we have proposed an integrated MAC and routing design to explore the capabilities provided by multiple channels and multiple interfaces in ad hoc networks. We defined 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 a limited number of channels. Based on the routing metric, we proposed a routing algorithm for path discovery, which considers all the major factors of a MCMI network in finding the minimum cost path. We also presented route maintenance schemes for adapting 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 the 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 minimize the effect of channel switching delay, balance the load, and enable fairness among neighboring nodes. In addition, we enhanced 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 a 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 very efficiently utilize the channel resources to significantly improve the network throughput in a multichannel multi-interface environment.

References

[1] A. Raniwala, K. Gopalan, and T. Chiueh, "Centralized algorithms for multichannel wireless mesh networks," *ACM Mobile Comput. Commun. Rev.*, vol. 8, no. 2, pp. 50–65, Apr. 2004.

[2] A. Raniwala and T. Chiueh, "Algorithms for an IEEE-802.11-based multichannel wireless mesh network," in *Proc. IEEE INFOCOM*, Mar. 2005, pp. 2223–2234.

[3] M. Alicherry, R. Bhatia, and L. Li, "Joint channel assignment and routing for throughput optimization in multiradio wireless mesh networks," in *Proc. ACM MobiCom*, Sep. 2005, pp. 58–72.

[4] A. Mishra, D. Agrawal, V. Shrivastava, S. Banerjee, and S. Ganguly, "Distributed channel management in uncoordinated wireless environments," in *Proc. ACM MobiCom*, Los Angeles, CA, Sep. 2006, pp. 170–181.

[5] K. N. Ramachandran, E. M. Belding, K. C. Almeroth, and M. M. Buddhikot, "Interference-aware channel assignment in multiradio wireless mesh networks," in *Proc. IEEE INFOCOM*, Apr. 2006, pp. 1–12.

[6] M. Kodialam and T. Nandagopal, "Characterizing the capacity region in multiradio, multichannel wireless mesh networks," in *Proc. ACM MobiCom*, Sep. 2005, pp. 73–87.

[7] A. Nasipuri and S. R. Das, "Multichannel CSMA with signal-power-based channel selection for multihop wireless networks," in *Proc. IEEE VTC*, Sep. 2000, pp. 211–218.

[8] N. Jain and S. R. Das, "Protocol with receiver-based channel selection for multihop wireless networks," in *Proc. IEEE IC^{3}N*, Oct. 2001, pp. 432–439.

[9] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, and J.-P. Sheu, "New multichannel MAC protocol with on-demand channel assignment for mobile ad hoc networks," in *Proc. I-SPAN*, Oct. 2000, pp. 232–237.

[10] J. So and N. H. Vaidya, "Multichannel MAC for ad hoc networks: Handling multichannel hidden terminals using a single transceiver," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, May 2004, pp. 222–233.

[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 *Proc. ACM MobiCom*, Sep. 2004, pp. 216–230.

[12] A. Tzamaloukas and J. J. Garcia-Luna-Aceves, "A receiver-initiated collision-avoidance protocol for multichannel networks," in *Proc. IEEE INFOCOM*, Apr. 2001, pp. 189–198.

[13] J. So and N. H. Vaidya, "A routing protocol for utilizing multiple channels in multihop wireless networks with a single transceiver," Univ. Illinois Urbana-Champaign, Champaign, IL, Tech. Rep., Oct. 2004.

[14] P. Kyasanur and N. Vaidya, "Routing and link-layer protocols for multichannel multi-interface ad hoc wireless networks," *SIGMOBILE Mobile Comput. Commun. Rev.*, 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 multichannel ad hoc wireless networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 6, pp. 1874–1887, Dec. 2009.

[16] S. Merlin, N. H. Vaidya, and M. Zorzi, "Resource allocation in multiradio multichannel multihop wireless networks," in *Proc. IEEE INFOCOM*, Apr. 2008, pp. 610–618.

[17] J. Shi, T. Salonidis, and E. W. Knightly, "Starvation mitigation through multichannel coordination in CSMA multihop wireless networks," in *Proc. 7th ACM Int. Symp. Mobile Ad Hoc Netw. Comput., MobiHoc*, 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)," IETF MANET Working Group (Draft 10), 2004.

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

[20] R. Draves, J. Padhye, and B. Zill, "Routing in multiradio, multihop wireless mesh networks," in *Proc. ACM MobiCom*, Sep. 2004, pp. 114–128.

[21] W.-H. Tarn and Y. C. Tseng, "Joint multichannel link layer and multipath routing design for wireless mesh networks," in *Proc. IEEE INFOCOM*, May 2007, pp. 2081–2089.

[22] D. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multihop wireless routing," in *Proc. ACM MobiCom*, Sep. 2003, pp. 134–146.

[23] H. Li, Y. Cheng, C. Zhou, and W. Zhuang, "Minimizing end-to-end delay: A novel routing metric for multiradio wireless mesh networks," in *Proc. IEEE INFOCOM*, Apr. 2009, pp. 46–54.

[24] L. Bao and J. Garcia-Luna-Aceves, "Hybrid channel access scheduling in ad hoc networks," in *Proc. IEEE ICNP*, Nov. 2002, pp. 46–57.

[25] J. Padhye, S. Agarwal, V. N. Padmanabhan, L. Qiu, A. Rao, and B. Zill, "Estimation of link interference in static multihop wireless networks," in *Proc. IMC*, Oct. 2005, p. 28.

[26] G. Bianchi and I. Tinnirello, "Kalman filter estimation of the number of competing terminals in an IEEE 802.11 network," in *Proc. IEEE INFOCOM*, Mar. 2003, pp. 844–852.

[27] X. Zeng, R. Bagrodia, and M. Gerla, "GLOMOSIM: A library for parallel simulation of large-scale wireless networks," in *Proc. 12th Workshop PADS*, May 1998, pp. 154–161.

[28] W. Navidi and T. Camp, "Stationary distributions for the random waypoint mobility model," *IEEE Trans. Mobile Comput.*, vol. 3, no. 1, pp. 99–108, Jan./Feb. 2004.

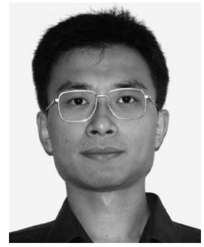
Jinhua Zhu received the Ph.D. degree in computer science and engineering from the University at Buffalo, Buffalo, NY, in 2005.

He is currently a Service Engineer with Ericsson Inc., Boulder, CO. His research interests include wireless sensor networks and cognitive radio.

Xin Wang (M'01) received the Ph.D. degree in electrical and computer engineering from Columbia University, New York, NY, in 2001.

From 2001 to 2003, she was a member of Technical Staff with Bell Labs Research, Lucent Technologies, Murray Hill, NJ, and an Assistant Professor with the Department of Computer Science and Engineering, University at Buffalo, Buffalo, NY. She is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY. Her research interests include wireless networks and communications and mobile and distributed computing.

Dr. Wang has served in numerous technical communities of top networking conferences.



Dahai Xu (S'01–M'05) received the Ph.D. degree from the University at Buffalo, Buffalo, NY, in 2005.

He spent two years as a Postdoctoral Research Associate with Princeton University, Princeton, NJ. He is currently a Research Staff with AT&T Labs Research, Florham Park, NJ. His research interests include Internet design, control and management, algorithm design and fast implementation, large-scale nonlinear network optimization, and secure communication in wireless ad hoc networks.

AUTHOR QUERIES

AUTHOR PLEASE ANSWER ALL QUERIES

AQ1 = T/R was defined as transmit-receive. Please check if this is correct. Otherwise, provide the corresponding definition.

AQ2 = WCETT was defined as weighted cumulative expected transmission time. Please check if this is correct. Otherwise, provide the corresponding definition.

AQ3 = RREQ, RRER, and RREP were defined as route request, route error, and route reply, respectively. Please check if these are correct. Otherwise, provide the corresponding definitions.

AQ4 = RES was defined as resume. Please check if this is correct. Otherwise, provide the corresponding definition.

AQ5 = CBR was defined as constant bit rate. Please check if this is correct. Otherwise, provide the corresponding definition.

END OF ALL QUERIES