

A Cross-layer Approach to Channel Assignment in Wireless Ad Hoc Networks

Michelle X. Gong · Scott F. Midkiff · Shiwen Mao

Published online: 4 December 2006
© Springer Science + Business Media, LLC 2006

Abstract To improve the capacity of wireless ad hoc networks by exploiting multiple available channels, we propose a distributed channel assignment protocol that is based on a cross-layer approach. By combining channel assignment with routing protocols, the proposed channel assignment protocol is shown to require fewer channels and exhibit lower communication, computation, and storage complexity than existing channel assignment schemes. A multi-channel MAC (MC-MAC) protocol that works with the proposed channel assignment protocol is also presented. We prove the correctness of the proposed channel assignment protocol. In addition, through a performance study, we show that the proposed protocol can substantially increase throughput and reduce delay in wireless ad hoc networks, compared to the IEEE 802.11 MAC protocol and an existing multi-channel scheme.

Keywords ad hoc routing protocol · cross-layer design · distributed channel assignment · multi-channel medium access control · wireless ad hoc networks

1 Introduction

Despite recent advances in wireless local area network (WLAN) technologies, today's WLANs still cannot offer the same data rates as their wired counterparts. The throughput problem is further aggravated in multi-hop wireless environments due to *intra-flow interference* introduced by adjacent nodes on the same path and *inter-flow interference* generated by nodes from neighboring paths. For instance, it has been shown that the maximum capacity that the IEEE 802.11 MAC can achieve for a chained network could be as low as one seventh of the link bandwidth [10].

All current IEEE 802.11 physical (PHY) standards divide the available frequency into several orthogonal channels, which can be used simultaneously within a neighborhood. Therefore, increasing capacity by exploiting multiple channels becomes particularly appealing. However, IEEE 802.11 WLANs that operate in ad hoc mode rarely use multiple channels simultaneously, partly because the IEEE 802.11 MAC is not designed to operate with multiple channels. For instance, an ad hoc network that is based on IEEE 802.11a technology uses only one out of 12 available orthogonal channels, wasting more than 90% of the potentially available spectrum.

Consequently, there has been substantial interest in channel assignment schemes that can achieve higher throughput by exploiting multiple available channels [8, 11, 14, 18]. The channel assignment problem has been shown to be \mathcal{NP} -complete and, thus, computationally intractable [3, 7]. In this paper, we propose an efficient distributed channel assignment protocol and an accompanying multi-channel MAC protocol. We design this channel assignment algorithm based on three

M. X. Gong
Cisco Systems, Inc., San Jose, CA 95134, USA

S. F. Midkiff
The Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061, USA

S. Mao (✉)
Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849, USA
e-mail: smao@ieee.org

design principles. *First*, to reduce complexity, channel assignment and routing should be jointly considered. Most existing multi-channel MAC protocols have two management entities: channel assignment and medium access control. We propose to bring the channel assignment entity to the routing layer. This “cross-layer” design approach is motivated by the fact that both the channel assignment algorithm and the ad hoc routing algorithm will be invoked when there is a change in the network topology. Exploiting this design principle can greatly reduce the complexity of channel assignment algorithms. In addition, the resulting multichannel MAC protocol can be simplified since it is relieved the channel assignment burden.

Second, channels should be assigned only to active nodes. This “on-demand” channel assignment principle is motivated by the fact that only active nodes need communication channels. Fewer channels may be required if this on-demand assignment principle is implemented. *Finally*, both collisions and interference should be taken into consideration. Two or more wireless nodes may generate *primary collisions* if they are one hop away from each other, while *secondary collisions* can be generated by nodes that are two hops away from each other [4]. Most existing channel assignment protocols consider only secondary collisions, since they are mainly designed to solve the “hidden terminal” problem [3, 7]. Nevertheless, primary collisions and interference are also important factors that adversely affect channel utilization and network capacity. To improve network performance, distinct channels should be assigned in such a way that collisions and interference can be avoided as much as possible.

We propose to combine channel assignment with a proactive ad hoc routing protocol called Optimized Link State Routing (OLSR) [5]. The proposed Channel Assignment OLSR (CA-OLSR) protocol assigns distinct channels to active nodes. Further, a multi-channel MAC protocol (MC-MAC) is incorporated with CA-OLSR to manage multi-channel medium access. A proactive routing protocol allows each node to have the complete up-to-date topology information of the network based on periodic exchanges of control messages. Therefore, proactive routing protocols can be closely coupled with channel assignment, without causing significant modifications to the protocols themselves.

As a result of the cross-layer approach, CA-OLSR exhibits significantly lower complexity than existing channel assignment protocols. Following the on-demand channel assignment principle, CA-OLSR is invoked only when there is a channel conflict, thus

further reducing control overhead. Unlike some channel assignment schemes that ignore a node’s one-hop neighbors [6, 7], CA-OLSR assigns distinct channels to active nodes within a k -hop neighborhood to avoid collisions and to mitigate interference.

The rest of the paper is organized as follows. We formulate the channel assignment problem in Section 2. We then describe the proposed CA-OLSR protocol and the multi-channel MAC protocol in Section 3, and provide a correctness proof in Section 4. A performance study with ns-2 simulations is presented in Section 5. Section 6 presents related work and Section 7 concludes this paper.

2 Problem formulation

A wireless ad hoc network can be modeled as a graph $G = \{V, E\}$, where V is the set of nodes and E is the set of edges that represent links. We assume that nodes use omnidirectional antennas and radio links are bidirectional. A link is assumed to exist between two nodes if and only if the two nodes are within each other’s radio range. Both primary and secondary collisions can be eliminated if nodes within a two-hop range of each other transmit on different orthogonal channels. The interference range is defined to be the k -hop neighborhood of a node.

Before we formulate the distributed channel assignment problem, we define $V_t \subset V$ to be the set of active transmitters and $V_r \subset V$ the set of active receivers. Note that V_t is determined by the underlying scheduling algorithm (e.g., IEEE 802.11 MAC or a multi-channel MAC). Let $v_{r,i} \in V_r$ be a particular receiver and $v_{t,j} \in V_t$ be a particular transmitter. Let $P(v_{t,j}, v_{r,i})$ denote the received power at node $v_{r,i}$, which is a function of the distance between nodes $v_{t,j}$ and $v_{r,i}$, the transmit power, and the channel condition (e.g., path loss). \mathcal{C} denotes the set of all available channels in the network. Table 1 summarizes the notation used.

If the number of available channels, i.e. $|\mathcal{C}|$, is sufficiently large, distributed channel assignment algorithms should assign distinct channels to any nodes within a k -hop neighborhood. However, in many cases, $|\mathcal{C}|$ may be less than the number of nodes in a k -hop neighborhood. Therefore, the objective of distributed channel assignment is to minimize the maximum number of nodes sharing the same channel with any designated node $v_{t,j} \in V_t$ among this node’s k -hop neighbors. We can formulate the channel assignment problem as follows.

Table 1 Notation

Symbol	Definition
C	Set of available channels
V	Set of nodes in the network
E	Set of edges that represent radio links
$v \in V$	A node in the network
$n_k(v)$	Number of k -hop neighbors sharing the same channel with node v
$V_t \subset V$	Set of active transmitters
$V_r \subset V$	Set of receivers
$v_{t,j} \in V_t$	A particular transmitter
$v_{r,i} \in V_r$	A particular receiver
$v_{t,T(i)}$	Desired transmitter
$P(v_{t,j}, v_{r,i})$	Power level of the received signal from transmitter $v_{t,j}$ to receiver $v_{r,i}$
$S(v_{t,T(i)}, v_{t,j})$	Relation between the channels used by $v_{t,T(i)}$ and $v_{t,j}$
β	SINR threshold for successful reception
P_N	Power level of additive white Gaussian noise

Minimize:

$$\max n_k(v_{t,j}), \quad \forall v_{t,j} \in V_t \tag{1}$$

subject to:

$$\frac{P(v_{t,T(i)}, v_{r,i})}{\sum_{v_{t,j} \in V_t \setminus v_{t,T(i)}} [P(v_{t,j}, v_{r,i}) \cdot S(v_{t,T(i)}, v_{t,j})] + P_N} \geq \beta, \quad \forall v_{r,i} \in V_r \tag{2}$$

where P_N is the power level of the additive white Gaussian noise (AWGN) noise and β is the minimum signal-to-interference-noise-ratio (SINR) required for a successful packet reception. The term $S(v_{t,T(i)}, v_{t,j})$ indicates the cross-correlation between the channels used by the desired transmitter $v_{t,T(i)}$ and transmitter $v_{t,j}$.

The SINR constraint implies that the cumulative interference generated by active transmitters sharing the same or interfering data channels as the designated transmitter should be less than a certain threshold to ensure that the receiver can decode the data packet successfully. Thus, a channel assignment protocol should distribute available channels within any k -hop neighborhood in such a way that the maximum number of transmitters sharing the same data channel is minimized. Meanwhile, the same set of channels should be re-used in such a way that the cumulative interference generated on any particular data channel is below a certain threshold.

3 Protocol description

Because the distributed channel assignment problem is shown to be \mathcal{NP} -complete [7], it is a great challenge to design practical channel assignment algorithms for general ad hoc networks. To this end, we explore efficient design of heuristic algorithms that can achieve near-optimal performance with low complexity. In this section, we describe the proposed CA-OLSR protocol in detail and provide a brief overview of the multi-channel MAC protocol. CA-OLSR seeks to use as many channel as possible within a neighborhood and to keep channel conflicts to a minimum. Meanwhile, the multi-channel MAC protocol guarantees that even when two neighboring nodes choose the same channel, contention and collisions on the channel are controlled and resolved.

3.1 Overview of Optimized Link State Routing (OLSR)

As the name suggests, OLSR is essentially a link state routing protocol with an optimized flooding method that can effectively reduce routing control overhead [5]. Specifically, OLSR minimizes the overhead of control packet flooding by using only selected neighbors, called multipoint relays (MPRs), to retransmit control messages. Each node selects its MPR set among its one hop neighbors in such a way that the set covers all the two-hop neighbors. For instance, the black nodes in Fig. 1 are MPRs selected by the node in the center. Once a node is selected as a multipoint relay, it not only retransmits routing control messages, but also serves as an intermediate node on routing paths.

In OLSR, two types of control messages are transmitted periodically: HELLO message and topology control (TC) message. HELLO messages permit each node to learn the topology of its neighbors up to two hops away. Based on this information, each node in the

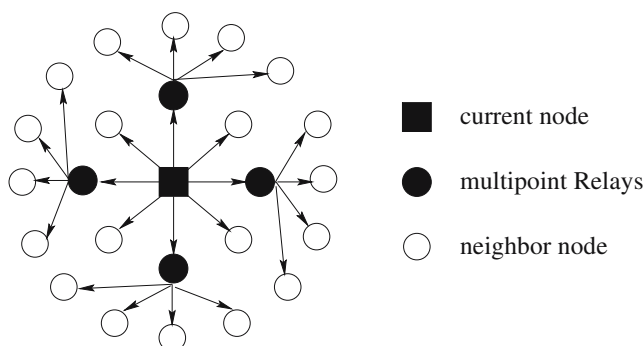


Figure 1 Illustration of multi-point relays

network independently selects its own set of MPRs that covers all the two-hop neighbors. TC messages are sent periodically by each node to declare its MPR selector set that consists of the list of neighbors who have selected the sender node as an MPR. TC messages are forwarded by nodes in the MPR set to the entire network. A TC message is larger than a HELLO message and is sent out less frequently than the HELLO message. A routing table is constructed at each node based on the information contained in received HELLO and TC messages. The calculation of the routing table can follow any standard link state algorithm. An example procedure is given in RFC 3626 to explain how the routing table is computed [5].

3.2 The Channel Assignment and Optimized Link State Routing (CA-OLSR) protocol

CA-OLSR combines distributed channel assignment with the OLSR routing protocol. We assume that there is one dedicated control channel and up to N equivalent data channels in the network. The dedicated control channel is shared by all the nodes in the network to exchange routing and MAC control messages. Each transmitter is assigned one data channel. The assigned data channel is then used by MC-MAC for data packet transmissions.

The basic ideas of CA-OLSR are to use routing control messages to exchange channel information and to assign distinct channels to active transmitters within a k -hop neighborhood. CA-OLSR identifies active nodes within a certain time period and then give them higher priority when there is a channel conflict. Because HELLO messages are sent more frequently and can carry channel information up to two hops away, we choose HELLO messages to carry channel information. The neighborhood size in CA-OLSR is defined to be $k = 2$, based on the assumption that interference range is about twice the radio range [11, 15].

With CA-OLSR, data channels are assigned according to the following procedure. During initialization, each node in the network randomly chooses a *NodeNumber* and a channel from a set of all available channels, denoted by \mathcal{C} . The random *NodeNumber* is used for resolving channel conflicts. The range of *NodeNumber* should be large enough such that the probability of two neighboring nodes choosing the same *NodeNumber* at the same time is extremely low. For instance, we choose the maximum range of a four-byte unsigned integer as the range of *NodeNumber*. Each time a node updates its data channel, it should re-generate a *NodeNumber*. When two neighboring nodes that choose the same data channel happen to have the same *NodeNum-*

Procedure *sendHello()*

```
HELLO.myChannelIndex = myChannel;
HELLO.NeighborNumber[0] = myNumber;
for i=1 to k
  HELLO.NeighborChannelIndex[i] = NeighborChannel[i];
  HELLO.NeighborNumber[i] = NeighborNumber[i];
endfor
Broadcast HELLO;
endprocedure
```

Procedure *recvHello(HELLO)*

```
(neighborChannels, NodeNumber) = getActiveNeighborInfo(HELLO);
Update the available channel set  $\mathcal{A}$ ;
channelConflict = FALSE;
if (Detect a channel conflict)
  if (activeNode == TRUE)
    if (activeMe == FALSE)
      channelConflict = TRUE;
    elseif (NodeNumber ≤ myNumber)
      channelConflict = TRUE;
    endif
  else
    if (activeMe == FALSE && NodeNumber ≤ myNumber)
      channelConflict = TRUE;
    endif
  endif
endif
if (channelConflict == TRUE)
  myChannel = randomChannel( $\mathcal{A}$ );
  myNumber = randomNumber();
endif
endprocedure
```

Figure 2 Procedures *sendHello()* and *recvHello(HELLO)*

ber, both of them should update their data channels and randomly choose another *NodeNumber*, as shown in Fig. 2. A node sends out a HELLO message that contains a list of its one-hop active neighbors, their channels, and their *NodeNumbers*. Nodes detect active neighbors by listening on the common control channel. Neighbors that exchange RTS and CTS control messages in one hello interval are considered active neighbors, which are then indicated in the HELLO messages sent in the next hello interval. Upon receiving HELLO messages from neighboring nodes, a node builds an available channel list, \mathcal{A} , by marking channels that are taken by active neighbors as unavailable.

If there is a channel conflict between the current node and an active neighbor, the current node should choose another channel from the available channel set \mathcal{A} . This is to ensure that active nodes have higher priority to obtain distinct channels than other nodes when the number of available channels is fewer than the number of nodes in the two-hop neighborhood. However, if there is a channel conflict between two active nodes, the node with the smaller *NodeNumber* retains its channel while the other node should mark the channel-in-conflict as unavailable and randomly pick a new channel from its updated available channel set \mathcal{A} . The same procedure applies when two inactive

nodes have a channel conflict. In general, active nodes have higher priority over in-active nodes. Within the same set of conflicting active or inactive nodes, the nodes with smaller *NodeNumber* have higher priority. Because *NodeNumbers* are randomly generated each time there is a channel update, CA-OLSR does not favor one node over another within the same set of conflicting active or inactive nodes. It is worth noting that the use of random *NodeNumbers* achieves fairness among the contending nodes, as compared with prior approaches that resolve channel conflicts based on static node IDs [6].

CA-OLSR has two main procedures that relate to channel assignment. These two procedures, `sendHello()` and `recvHello(HELLO)`, are summarized in Fig. 2. In the `recvHello(HELLO)` procedure, the `getActiveNeighborInfo(HELLO)` function retrieves channel information from HELLO messages received from neighboring nodes, while the `randomChannel(A)` function returns a channel index that is randomly chosen from the available channel set \mathcal{A} . Note that the channel assignment procedure in CA-OLSR is invoked “on-demand,” only when a channel conflict occurs after a topology change. Further, CA-OLSR seeks to assign distinct channels to active nodes in the two-hop neighborhood to minimize collisions as well as to mitigate interference.

3.3 Overview of the Multi-Channel MAC protocol

Most current multi-channel MAC protocols have two functionalities: channel assignment and medium access control [8, 11, 18]. Because the channel assignment is performed by CA-OLSR, our Multi-Channel MAC (MC-MAC) protocol needs to manage only medium access control on multiple data channels. As a result, the design of MAC protocol is significantly simplified. A second advantage of our approach, i.e. the separation of channel assignment and medium access control, is that it enables optimization of different modules separately. For instance, channel assignment can be combined with different reactive or proactive routing protocols. The MAC protocol can also be designed independently without the knowledge of how channels are assigned to individual nodes. In addition, the separation of functions makes it possible to design backward compatible and practical multi-channel MAC protocols.

MC-MAC is a transmitter-based protocol. It is assumed that all nodes in the network share the same common control channel and each node is equipped with two half-duplex transceivers. One transceiver at each node listens on the common control channel all the time, whereas the other transceiver can switch

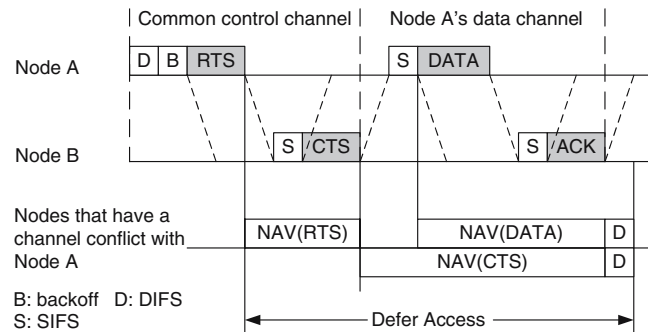


Figure 3 Four-way handshake procedure of MC-MAC

from one channel to another. Nodes are assigned data channels by the CA-OLSR routing protocol. When a node is ready to transmit, it first informs the destination node of its assigned data channel. As shown in Fig. 3, when the sender Node A intends to transmit, it first uses the control channel to broadcast a request-to-send (RTS) message, carrying its own data channel index, c_A . Upon receiving the RTS message, the destination Node B uses the control channel to return a clear-to-send (CTS) message carrying c_A and switches its receiving channel to c_A . After Node A receives the CTS message on the control channel, the other transceiver switches to the confirmed data channel c_A and starts data transmission. Neighboring nodes that overhear the RTS/CTS exchange but do not share the same data channel with Node A should defer only for the duration of the control message transmission. If a node is assigned the same data channel c_A , two situations may happen, as illustrated in Fig. 3.

- If the node overhears a CTS message, it should defer from using the data channel c_A until the end of the data transmission to avoid causing a collision at the receiver.
- The node that overhears only an RTS message, but not a CTS message, should first defer from using the control channel only for the duration of the control packet transmission. Then, it performs carrier sensing on its own data channel c_A . If the carrier is busy, which means that the transmitting node has successfully acquired the medium, the node should defer for the duration of the data packet transmission. However, if the carrier is not busy, the node can start to contend for the control channel immediately.

After the data transmission, the sender listens on the data channel until an ACK is received or a timeout occurs. If a node receives an RTS while its data transceiver is busy communicating with another node, it replies with a Negative CTS (NCTS) on the control

channel to the sender of the RTS. Upon receiving an NCTS message, the initiator of the RTS knows that a collision has not occurred and thus is not obliged to increase its contention window nor to backoff. Because each node in the network is equipped with two half-duplex transceivers, nodes can transmit and receive on different channels simultaneously. The most significant difference between our Multi-Channel MAC (MC-MAC) protocol and other multi-channel MAC protocols is that MC-MAC only needs to perform medium access control, whereas other multi-channel MAC protocols must also perform channel assignment.

4 Correctness of CA-OLSR

Under the assumption that the number of available channels is sufficiently large and channel information is updated more often than topology changes, we show that CA-OLSR can assign distinct channels to any active node within a two-hop range. For the purpose of analysis only, CA-OLSR is assumed to operate under the following conditions. (1) The number of available channels is more than the number of active nodes within a two-hop range at any time. (2) Each node chooses a unique *NodeNumber* in the network at any time. Note that CA-OLSR can easily resolve *NodeNumber* conflicts by allowing nodes that are involved in *NodeNumber* conflicts choose other random numbers. Because a *NodeNumber* conflict is much easier to resolve than a channel conflict, to simplify the analysis, we assume that *NodeNumbers* are collision-free. (3) All HELLO messages transmitted over a radio link are received correctly in the same hello interval. (4) Topology and channel changes can be conveyed to all nodes within a two-hop neighborhood before the next topology change. This could be achieved by choosing an appropriate hello interval according to the current mobility level [5].

Theorem 1 *Within a finite time after a topology change, CA-OLSR can assign distinct channels to any active nodes within a two-hop neighborhood.*

To prove the above statement, we need to show that the following statements hold true.

1. CA-OLSR is dead-lock free and live-lock free.
2. Within a finite amount of time after a topology change, all nodes must have consistent and up-to-date topology and channel information about nodes within its two-hop neighborhood.

3. If the information at each node is consistent and up-to-date, CA-OLSR can assign channels correctly, such that no two or more active nodes in the two-hop neighborhood share the same data channel.

Lemma 1 *CA-OLSR is dead-lock free and live-lock free. In addition, the channel updating procedure will take at most $l - 1$ steps, where $l \geq 2$ is the number of active nodes that are involved in a channel conflict.*

Proof The proof of the dead-lock free property of CA-OLSR is straight forward. In CA-OLSR, each node in the network is assigned only one channel. However, for deadlocks to occur, a node must hold at least one resource (i.e. a channel) before it requesting another resource. Since CA-OLSR assigns only one channel to each node, there is no deadlock.

Next, we prove that CA-OLSR is also live-lock free. A live-lock may happen when two or more nodes continually update their channels in response to changes in other nodes. The result is that none of the nodes will stop updating their channels and the channel assignment algorithm may never converge. In CA-OLSR, an active node may update its data channel only when there is a channel conflict between this node and at least one of its one-hop or two-hop active neighboring nodes. Three scenarios may occur where at least one node in the network needs to switch its data channel.

Case 1 Assuming that there are exactly two active nodes that choose the same data channel, one node must have a *NodeNumber* that is smaller than that of the other node because of the assumption that *NodeNumbers* are unique within the network. Then, the node with a smaller *NodeNumber* keeps its data channel while the node with a larger *NodeNumber* updates its own data channel to avoid the channel conflict. In this two-node case, no live-lock will occur.

Case 2 Assuming that there are $l > 2$ active nodes that pick the same data channel, there must be a node that has the smallest *NodeNumber* among the l active nodes. The node with the smallest *NodeNumber* keeps its data channel, while each of the rest $l - 1$ nodes pick a new data channel for itself. Note that because the original data channel has been marked as unavailable at each of the $l - 1$ nodes, all of them pick a data channel that is different from the original one. In the worst-case situation where all $l - 1$ nodes pick the same data channel again, the node that has the smallest *NodeNumber* among the $l - 1$ nodes can retain its data channel while the other $l - 2$ nodes randomly pick a new data channel from the available channel set \mathcal{A} . The

available channel set \mathcal{A} at each node is updated at each step. The channel update procedure continues until all l nodes have distinct data channels. In the worst case, this takes $l - 1$ steps. Because at every step the node that can retain its original channel is determined by its *NodeNumber*, which is assumed to be unique in the network, no live-lock can occur either.

Case 3 In the case that there are two or more different channel conflicts that involve a total of l active nodes, the same upper bound still holds. First, because each node is assigned only one channel at any time, different channel conflicts always involve different nodes. Thus, different channel conflicts can be resolved independently in parallel. The worst-case scenario occurs when at the second step, all remaining nodes pick the same data channel. Then, this situation degenerates into Case 2.

As can be seen from the above three cases, no deadlock and live-lock can occur in CA-OLSR. In addition, the channel update procedure takes at most $l - 1$ steps, where l is the number of active nodes that have a channel conflict. ■

Corollary 1.1 *The value of the Hello interval should be $3(l - 1)$ times smaller than the topology change interval, where l is the maximum number of neighbors that can pick the same channel at the same time.*

Proof After updating its own data channel, a node v should send the updated topology information and channel information to its one-hop neighbors in either the current hello interval, h_0 , or the next hello interval, h_1 . To get the upper bound, we assume that the updated HELLO message is always sent out in the second hello interval h_1 . Upon receiving the updated HELLO message, one-hop neighbors of the current node convey the updated channel information to two-hop neighbors by sending their HELLO messages in the next hello interval. Therefore, a channel update may take up to three hello intervals to reach all the nodes within a two-hop range.

The proof of Lemma 1 shows that it takes at most $l - 1$ steps to resolve channel conflicts that involves l active nodes. To complete the channel update procedure, the updated channel information may need to propagate to all nodes in the two-hop neighborhood at each step. Since each step of the channel update procedure takes up to three hello intervals, the whole procedure takes up to $3(l - 1)$ hello intervals. Because the channel update procedure has to be completed before the next topology change, the hello interval should be $3(l - 1)$ times shorter than the topology change interval. ■

Lemma 2 *Within a finite amount of time after a topology change, all nodes have consistent and up-to-date topology and channel information about nodes within its two-hop neighborhood.*

Proof Consistent and up-to-date topology and channel information implies that a node knows the most recent topology change and all the recent channel changes among its one-hop and two-hop neighbors.

We know from Lemma 1 that CA-OLSR does not deadlock or live-lock. Specifically, the channel update procedure terminates after at most $l - 1$ steps. Further, it is assumed that the next topology change will not interfere with the channel update procedure invoked by the current topology change. Therefore, within a finite time, the channel updating procedure terminates and all nodes eventually stop updating their available channel set \mathcal{A} and their own data channels. All HELLO messages transmitted over a radio link are assumed to be received correctly in the same hello interval. Thus, at most three hello intervals after the channel selection algorithm terminates, HELLO messages can propagate to any node's two-hop neighbors. Therefore, within a finite amount of time after a topology change, all nodes must have consistent and up-to-date topology and channel information about nodes within their two-hop neighborhood. ■

Lemma 3 *When the channel update procedure terminates, no two active nodes in the two-hop neighborhood share the same data channel.*

Proof The channel update procedure terminates when nodes do not make any new channel update. Since the topology and channel information in each node are consistent and up-to-date, as stated in Lemma 2, the channel update procedure terminates based on the correct topology and channel information. Because the channel update procedure terminates only when there is no channel conflict within any two-hop neighborhood, no two active nodes in the two-hop neighborhood share the same data channel. ■

Based on the proven lemmas, we conclude that within a finite time after a topology change, CA-OLSR can assign distinct channels to any active nodes within a two-hop neighborhood. Thus, Theorem 1 is proved.

5 Performance evaluation

In this section, we present a performance study of the proposed protocol using ns-2 simulations [17].

We first compare CA-OLSR with an existing channel assignment protocol, i.e. the channel assignment scheme (CAS) [6]. We choose CAS because it is one of the distributed channel assignment protocols that can operate in mobile ad hoc networks. Additionally, unlike other channel assignment schemes that were proposed to work with the IEEE 802.11 MAC protocol [9, 11, 14], CAS can work with our proposed MC-MAC protocol. In addition, we demonstrate the capacity improvement achieved by CA-OLSR combined with MC-MAC over two benchmark schemes: (1) the single-channel IEEE 802.11 MAC and (2) MC-MAC with randomly assigned channels. Finally, we present several important observations made from our simulation studies.

We assume that 64 wireless nodes are placed randomly in a squared area. It is assumed that each node is equipped with two transceivers. All nodes in the network share the same common control channel. The physical bandwidth for each data channel and the control channel is set to 2 Mb/s [15]. For routing, the standard OLSR is used with the IEEE 802.11 MAC, while CA-OLSR is used with MC-MAC. Six UDP flows are generated in the network. Each UDP flow has an offered load ranging from 40 to 1,000 Kb/s. Most current wireless LAN cards have a channel switch delay of 40 to 80 μ s [1]. We assume a channel switch delay of 80 μ s. Two mobile ad hoc networks are simulated. The first network is an 800 m \times 800 m *dense* network and the second network is a 1,600 m \times 1,600 m *sparse* network. Mobile nodes move randomly according to the random waypoint mobility model, where node speed ranges from 4 to 5 m/s, and the maximum pause time is 5 s [15].

Table 2 Simulation settings

Simulation parameter	Value
Number of nodes	64
Radio range	250 m
Interference range	550 m
Physical channel bandwidth	2 Mb/s
Path loss model	Two-ray ground
Mobility model	speed: 4 to 5 m/s pause time: 5 s
Dense network	800 m \times 800 m
Sparse network	1,600 m \times 1,600 m
Available control channel	1
Available data channels	6 or 12
Channel switch delay	80 μ s
Warm-up period	2,000 s
Effective simulation time	4,000 s

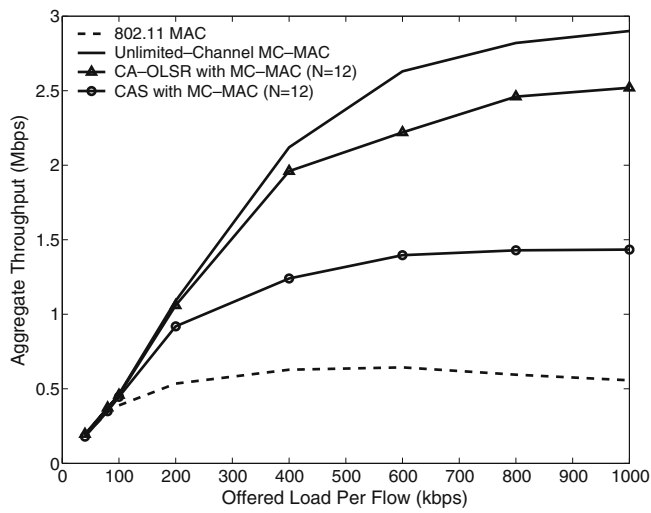
Although we have also studied CA-OLSR at higher speeds, the results are not significantly different and are omitted for brevity. The network density has a much greater impact on the performance than node mobility. Similar observations have also been made by Bahl et al. in [1]. Table 2 summarizes the simulation setting for the results reported in this paper.

5.1 Comparison with an existing channel assignment scheme

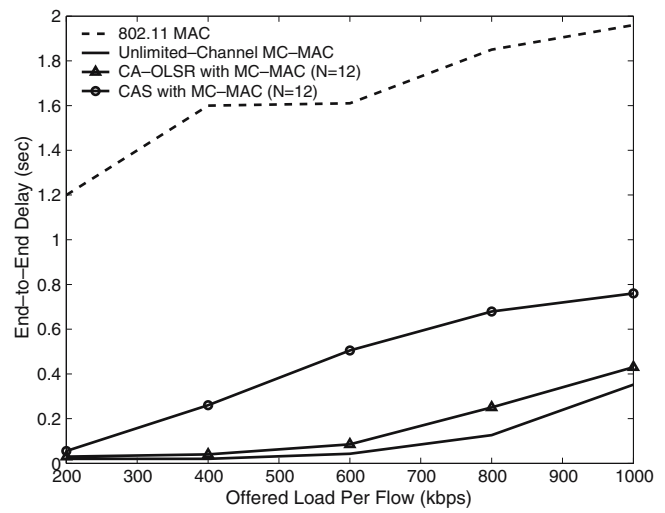
We first compare CA-OLSR with an existing channel assignment scheme (CAS) [6]. Simulation results for the IEEE 802.11 MAC and a scheme that utilizes unlimited number of data channels, are also provided as benchmarks. In the unlimited data channel scheme, each node has its own unique data channel in the network and a common control channel is shared by all nodes in the network. Therefore this gives an upper bound for the performance of any practical channel assignment scheme.

Proposed by Garcia-Luna-Aceves and Raju [6], CAS assigns distinct channels, or codes, to a node and its two-hop neighbors. Each node sends out code assignment messages (CAM) to its one-hop neighbors. CAMs are sent in three conditions: (1) when a new node comes up, (2) when a node detects a change of code by any of its one-hop neighbors, and (3) when a node finds that one of its one-hop neighbors is no longer active. After receiving the CAM, each neighbor needs to acknowledge the message individually. Each CAM contains: (1) the address and the code of the node that sent the CAM, (2) the addresses and the codes of the node's one-hop neighbors, (3) acknowledgements to earlier CAMs, and (4) a response list of zero or more nodes which need to send an ACK for this CAM. The communication complexity of CAS is $O(d^2 \cdot |V|)$, where d is the maximum number of one-hop neighbors for any node and $|V|$ is the total number of nodes in the network. In contrast, CA-OLSR has constant complexity, $O(1)$, since channel information is carried in routing control messages.

Figures 4a and 5a show that CAS performs worse than CA-OLSR in both dense and sparse networks in terms of throughput. The performance gap between CAS and CA-OLSR is more significant in a dense network because the communication overhead of CAS, i.e. $O(d^2 \cdot |V|)$, is higher in a denser network. Figures 4b and 5b show that the delay performance of CAS is better than that of the IEEE 802.11 MAC protocol. By utilizing multiple channels, CAS can effectively reduce



(a) Aggregate system throughput.



(b) Average end-to-end packet delay.

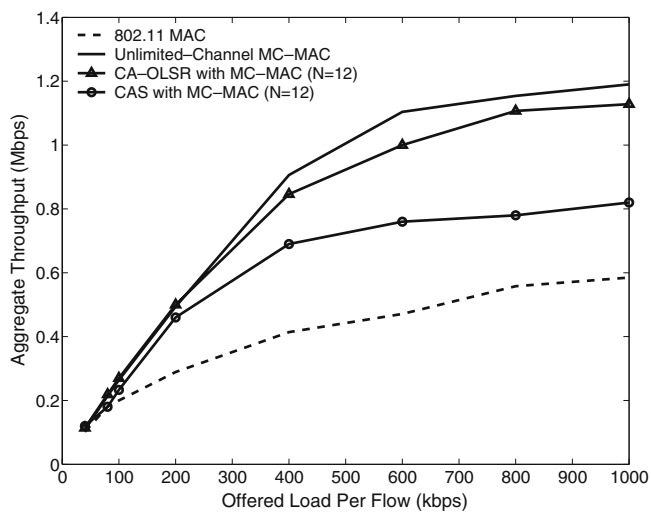
Figure 4 CA-OLSR vs CAS in a dense network

collisions and contention in the network. Thus, the end-to-end delay suffered by a data packet is reduced as compared to the original IEEE 802.11 MAC. However, the delay achieved by CAS is still not as good as that of CA-OLSR, which is more efficient in utilizing multiple channels with a lower control overhead.

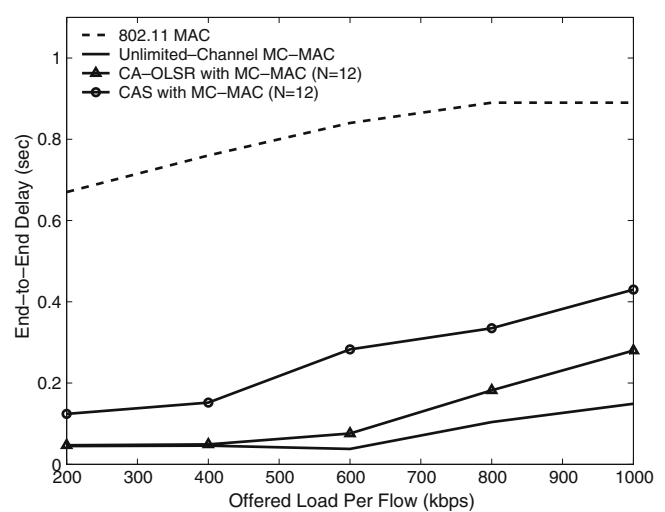
5.2 Comparison with the random channel assignment scheme and IEEE 802.11 MAC

We then compare CA-OLSR with the random channel assignment scheme and the IEEE 802.11 MAC protocol. In the random-OLSR algorithm, a node determines its own channel index based on its MAC address.

Therefore, this random scheme has constant communication complexity, $O(1)$. Figure 6a shows that CA-OLSR always performs better in terms of throughput than the random-OLSR scheme in a dense network, given an equal number of channels. The performance gap between the two increases as the number of available channels decreases. Note that CA-OLSR combined with MC-MAC can have a throughput up to four times higher than that of the IEEE 802.11 MAC. Figure 6b shows that the end-to-end delay increases for all schemes when the data rate increases. However, multi-channel schemes that utilize 12 data channels have much lower delay than the original 802.11 MAC scheme.

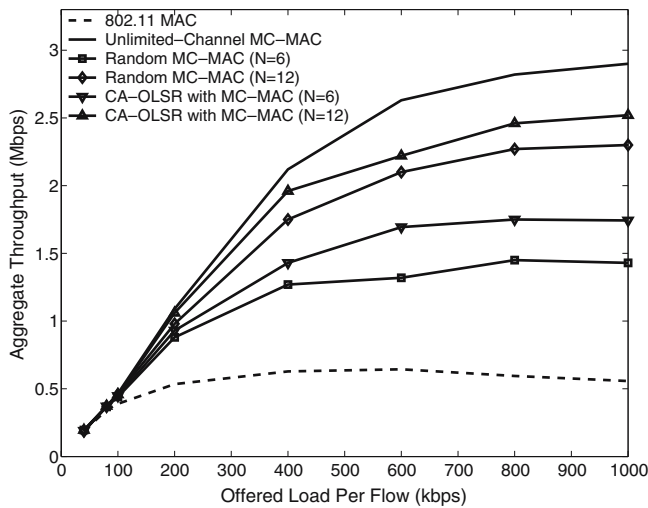


(a) Aggregate system throughput.

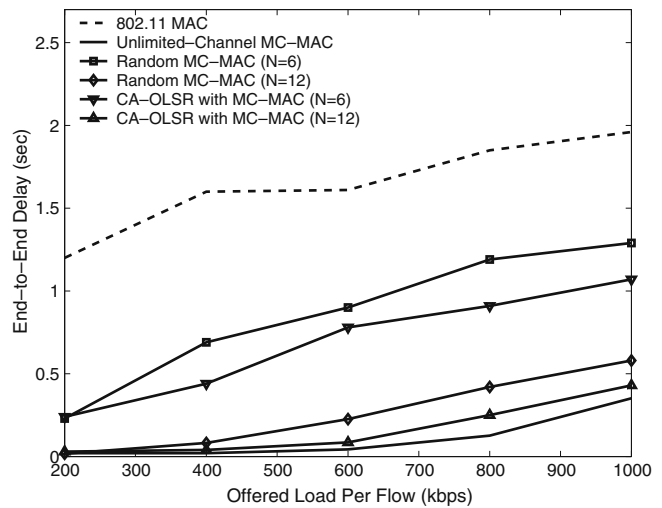


(b) Average end-to-end packet delay.

Figure 5 CA-OLSR vs CAS in a sparse network



(a) Aggregate system throughput.



(b) Average end-to-end packet delay.

Figure 6 Performance of CA-OLSR in a dense network

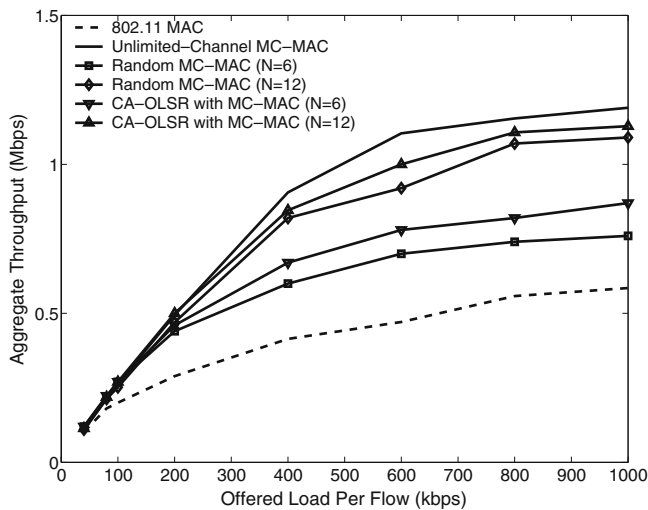
Figures 7a and 7b show the throughput and delay performance for the schemes in a sparse network. The performance gaps between different schemes are not as large as those in the dense network, although CA-OLSR still outperforms the random-OLSR scheme with a clear margin. This is because in a sparse network, each node has fewer neighbors. Interference and collisions generated by neighboring nodes in a sparse network is lower compared to those in a dense network. Therefore, the performance gain of multi-channel MAC schemes in a sparse network is not as significant as that in a dense network. Figure 8 shows the performance improvement with increased number of available data channels. The performance gap between

CA-OLSR and random-OLSR decreases slightly when the number of available channels increases.

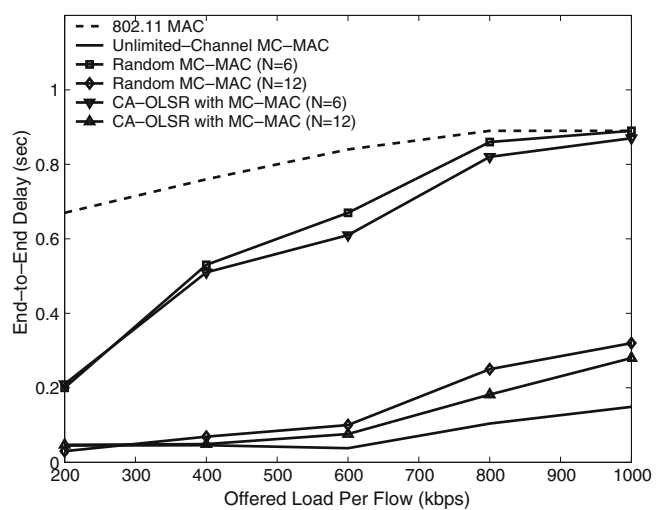
5.3 Observations

Based on the presented simulation results, we make the following observations.

- Communication overhead has a profound impact on the performance of distributed channel assignment protocols. Both CA-OLSR and CAS seek to assign distinctive channels to nodes in a two-hop neighborhood. However, because CA-OLSR has lower



(a) Aggregate system throughput.



(b) Average end-to-end packet delay.

Figure 7 Performance of CA-OLSR in a sparse network

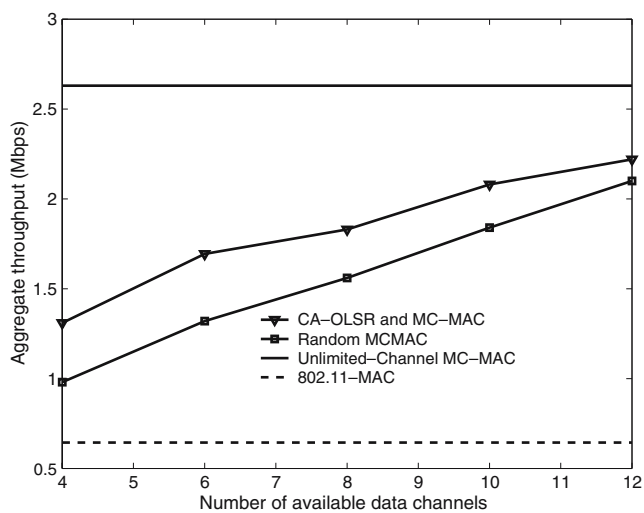


Figure 8 Impact of the number of channels on the aggregate throughput

communication overhead, it has much better performance than CAS, especially in a dense network.

- As expected, if the control overhead is not prohibitively high, utilizing multiple channels always gives better performance than the IEEE 802.11 MAC scheme. The reason is that utilizing multiple channels increases the capacity available in wireless ad hoc networks.
- In all simulated scenarios, the performance gap between the unlimited channel scheme and CA-OLSR with 12 data channels is not significant. Thus, we conjecture that due to the negligible interference generated by distant nodes, a large number of data channels is not necessary to achieve most of the benefits.
- The performance gap between CA-OLSR and the random scheme decreases when the number of available channels increases. We showed in Section 4 that CA-OLSR can assign distinct channels to any active node in a two-hop neighborhood. The random scheme merely tries to utilize all available channels without intelligent channel assignment. However, when there is a large number of available channels, collisions may be infrequent even if channels are assigned randomly.
- The performance gain achieved by CA-OLSR combined with MC-MAC over the IEEE 802.11 MAC scheme is not in proportion to the number of channels utilized. For instance, for MC-MAC with 12 available data channels, the throughput gain is about five times of that achieved by the IEEE 802.11 MAC, rather than 12 times in the idea case. This is because for MC-MAC, nodes have to exchange RTS/CTS messages on the control channel

before they can transmit on a data channel. With an increase in the number of data channels, the time period allocated to the common control channel also increases because the number of collisions on the common control channel increases as more nodes try to transmit in parallel. Therefore, the time period a transceiver can spend on transmitting data packets is reduced, which degrades the performance advantage of CA-OLSR. In addition, the channel switch delay is non-negligible, which further degrades the performance.

6 Related work

The prior work relevant to CA-OLSR and MC-MAC can be divided into two categories: distributed channel assignment and multi-channel MAC protocols, as discussed in the following.

6.1 Related work on distributed channel assignment

Hu's pioneering work [7] examined distributed code assignment for CDMA packet radio networks. The transmitter-based code assignment problem is transformed into an \mathcal{NP} -complete graph coloring problem. The heuristic algorithms proposed in [7] have high computation complexity and communication overhead. Moreover, the schemes do not consider the case when the number of codes is limited and perfect assignment is not possible.

Garcia-Luna-Aceves and Raju [6] describe a distributed code assignment scheme (CAS) that works in a mobile ad hoc network. As shown in Section 5, CAS assigns distinct channels to a node and its two-hop neighbors. If the number of available channels is at least $d(d-1)+2$, where d is the maximum number of neighbors for any node, it is shown that there will be no channel conflicts after the algorithm converges. However, this algorithm does not consider the case when the number of available channels is less than $d(d-1)+2$.

There are several recent proposals for routing protocols that are suitable for multi-hop multi-channel wireless mesh networks [9, 11, 13, 14]. The approach taken by most of these proposals is to combine routing with intelligent multi-channel assignment, such that channel utilization is maximized and the system performance can be substantially improved. However, because the routing protocols are the focus of these papers, the performance of the channel assignment schemes has not been explicitly studied [9, 11, 13]. Further, some channel assignment protocols have very high time

Table 3 Comparison of CA-OLSR to existing algorithms

Protocols	No. of Channels	Communication	Computation	Storage
Centralized greedy algorithm [7]	$d(d-1)+1$	N/A	$d^2 \cdot V $	$d^2 \cdot V $
Distributed channel assignment [6]	$d(d-1)+2$	$d^2 \cdot V $	d^2	d^2
Random scheme	N/A	$O(1)$	$O(1)$	$O(1)$
CA-OLSR	$d_a(d_a-1)+1$	$O(1)$	$O(d_a^2)$	$O(1)$

complexity, i.e. $O(Kn^3 \log m + m^2)$, where n is the total number of nodes in the network, m is the total number of radio connections in the network, and K is the minimum number of neighbors for any node [14].

Table 3 compares several heuristic algorithms in terms of their communication, computation, and storage complexity. In the table, d_a is the maximum number of active one-hop neighbors for any node, $|V|$ is the total number of nodes in the network, and k is the interference range. For CA-OLSR, the channel assignment information is piggybacked in routing messages, and channel assignments for each neighbor node are saved along with each entry in the neighbor table. Therefore, CA-OLSR has low communication overhead and storage overhead.

6.2 Related work on multi-channel MAC protocols

A significant body of prior work examines the benefits of utilizing multiple channels. According to the number of required transceivers or network interface cards (NICs), existing multi-channel MAC protocols can be divided into three categories: (1) multiple transceivers, such as DCA [18], (2) one transmitter and two or more receivers [2, 8], and (3) a single half-duplex transceiver,

such as MMAC [12], RICH-DP [16], and SSCH [1]. Note that MMAC, RICH-DP, and SSCH require node synchronization to some extent. Our MC-MAC protocol needs two transceivers and it does not require any node synchronization.

There is a fundamental tradeoff between the hardware complexity and the system performance. The multi-channel MAC schemes that require only a single transceiver can yield at least a factor of two improvement over single-channel MAC protocols [1, 12]. Our simulation results show that an improvement up to a factor of five can be achieved by our CA-OLSR and MC-MAC. A multi-channel MAC scheme that utilizes multiple NICs may yield better performance at the cost of higher hardware complexity. However, increasing the number of transceivers per node may cause excessive energy consumption and significant interference [11].

Table 4 summarizes some important features of existing multi-channel MAC protocols. There is no general rule as to which scheme is better than another. Simpler schemes with less hardware requirements are often easy to implement, whereas complex schemes with greater hardware requirements often yield better performance.

Table 4 Comparison of MC-MAC to existing multi-channel MAC protocols

Protocols	Medium access	Channel selection	Hardware requirement	Synchro. required
Dynamic Channel Assignment (DCA) [18]	CSMA/CA	Per packet	2 transceivers	No
Multi-channel MAC (MMAC) [12]	CSMA/CA	Per beacon interval	1 transceiver	Yes
Multi-channel CSMA [8]	CSMA/CA	Per packet	1 transmitter multiple receivers	No
Receiver-Initiated Channel-Hop with Dual Polling (RICH-DP) [16]	Channel hopping	Hopping sequence	1 transceiver	Yes
Slotted Seeded Channel Hopping (SSCH) [1]	Channel hopping and CSMA/CA	Hopping sequence	1 transceiver	Yes
Common Control Channel MMAC (CCC-MMAC) [2]	CSMA/CA	Per Mesh TXOP	1 transmitter 2 receivers	No
Multi-Channel MAC (MC-MAC)	CSMA/CA	Per route change	2 transceivers	No

7 Conclusions

In this paper, we presented a distributed channel assignment scheme that is based on a cross-layer approach. The proposed CA-OLSR protocol exhibits significantly lower communication, computation, and storage complexity than existing channel assignment schemes. We proved the correctness of the CA-OLSR protocol. Our simulation results show that CA-OLSR combined with MC-MAC can achieve up to a factor of five improvement in throughput over the IEEE 802.11 MAC protocol.

Acknowledgements This research was supported in part by a National Science Foundation Integrated Graduate Education and Research Training (IGERT) grant (award DGE-9987586).

References

- Bahl P, Chandra R, Dunagan J (2004) SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks. In: Proc. ACM MobiCom, Tokyo, Japan, pp 216–230
- Benveniste M (2005) CCC MMAC protocol framework and optional features. IEEE 802.11–05/0880r1
- Bertossi A, Bonuccelli M (1995) Code assignments for hidden terminal interference avoidance in multihop packet radio networks. IEEE/ACM Trans Netw 3(4):441–449
- Bharghavan V, Demers A, Shenker S, Zhang L (1994) MACAW: A media access protocol for wireless LANs. In: Proc. SIGCOMM. London, UK, pp 212–225
- Clausen T, Jacquet P (2003) Optimized link state routing protocol. IETF RFC 3626
- Garcia-Luna-Aceves J, Raju J (1997) Distributed assignment of codes for multihop packet-radio networks. In: Proc. MILCOM. Monterey, CA, pp 450–454
- Hu L (1993) Distributed code assignments for CDMA packet radio networks. IEEE/ACM Trans Netw 1(6):668–677
- Jain N, Das S, Nasipuri A (2001) A multichannel CSMA MAC protocol with receiver-based channel selection for multihop wireless networks. In: Proc. IEEE IC3N. Scottsdale, AZ, pp 432–439
- Kyasanur P, Vaidya N (2005) Routing and interface assignment in multi-channel multi-interface wireless networks. In: Proc. IEEE WCNC. Las Vegas, NV, pp 2051–2056
- Li J, Blake C, Coute DD, Lee H, Morris R (2001) Capacity of ad hoc wireless networks. In: Proc. ACM MobiCom. Rome, Italy, pp 61–69
- Raniwala A, Chiueh T (2005) Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. In: Proc. IEEE INFOCOM. Miami, FL, pp 2223–2234
- So J, Vaidya N (2004) Multi-channel MAC for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver. In: Proc. ACM MobiHoc. Tokyo, Japan, pp 222–233
- So J, Vaidya N (2005) Routing and channel assignment in multi-channel multi-hop wireless networks with single-NIC devices. Technical report, University of Illinois at Urbana-Champaign
- Tang J, Xue G, Zhang W (2005) Interference-aware topology control and QoS routing in multi-channel wireless mesh networks. In: Proc. ACM MobiHoc. Urbana-Champaign, IL, pp 68–77
- The CMU Monarch Project, Wireless and mobility extension to ns
- Tzamaloukas A, Garcia-Luna-Aceves JJ (2001) A receiver-initiated collision-avoidance protocol for multi-channel networks. In: Proc. IEEE INFOCOM. pp 189–198
- VINT Group, UCB/LBNL/VINT network simulator ns (version 2)
- Wu S-L, Lin C-Y, Tseng Y-C, Sheu J-P (2000) A new multi-channel MAC protocol with on-demand channel assignment for multihop mobile ad hoc networks. In: Proc. International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN). pp 232–237



Michelle Xiaohong Gong received her Ph.D. in Electrical Engineering from Virginia Tech in 2005. Currently, she is a member of the CTO group within the Wireless Networking Business Unit of Cisco Systems. She is actively involved as a voting member in the IEEE 802.11 standards body, and in particular the mesh networking Task Group. She has authored four pending US patents, two book chapters, and numerous papers in the areas of wireless communications and wireless networks. Her research interests include performance analysis and algorithm design for wireless networks, with a focus on medium access control and wireless routing protocols in wireless mesh networks.



Scott F. Midkiff received the B.S.E. and Ph.D. degrees from Duke University, Durham, NC, and the M.S. degree from Stanford University, Stanford, CA, all in electrical engineering. He worked at Bell Laboratories and held a visiting position at Carnegie Mellon University, Pittsburgh, PA. In 1986, he joined the Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg, where he is now a Professor. His research interests include system issues in wireless and ad hoc networks, network services for pervasive computing, and performance modeling of mobile ad hoc networks.



Shiwen Mao received the B.S. and the M.S. degree from Tsinghua University, Beijing, P.R. China in 1994 and 1997, respectively, both in Electrical Engineering. He received the M.S. degree in System Engineering and the Ph.D. degree in Electrical and Computer Engineering from Polytechnic University, Brooklyn, NY, in 2000 and 2004, respectively.

He was a Research Member at the IBM China Research

Lab, Beijing from 1997 to 1998. In the summer of 2001, he was a research intern at Avaya Labs-Research, Holmdel, NJ. He has been a Research Scientist in the Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA from December 2003 to April 2006. Currently, he is an Assistant Professor in the Department of Electrical and Computer Engineering at Auburn University, Auburn, AL.

Dr. Mao's research interests include cross-layer design and optimization in multi-hop wireless networks, as well as multimedia communications. He is a co-recipient of the 2004 *IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems*. He is the co-author of a textbook, *TCP/IP Essentials: A Lab-Based Approach* (Cambridge University Press, 2004).