Available online at www.sciencedirect.com
ScienceDirect
Ad Hoc Networks 7 (2009) 63–78
Ad Hoc Networks
www.elsevier.com/locate/adhoc

On-demand routing and channel assignment in multi-channel mobile ad hoc networks ☆

Michelle X. Gong [a], Scott F. Midkiff [b], Shiwen Mao [c,*]

[a] Corporate Technology Lab, Intel Corporation, Santa Clara, CA 95054, USA
[b] The Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061, USA
[c] Department of Electrical and Computer Engineering, Auburn University, 200 Broun Hall, Auburn, AL 36849, USA

Received 8 December 2006; received in revised form 6 April 2007; accepted 27 November 2007
Available online 5 December 2007

Abstract

The capacity of mobile ad hoc networks is constrained by the *intra-flow* interference introduced by adjacent nodes on the same path, and *inter-flow* interference generated by nodes from neighboring paths. By assigning orthogonal channels to neighboring nodes, one can minimize both types of interferences and allow concurrent transmissions within the neighborhood, thus improving the throughput and delay performance of the ad hoc network. In this paper, we present three novel distributed channel assignment protocols for multi-channel mobile ad hoc networks. The proposed protocols combine channel assignment with distributed on-demand routing, and only assign channels to active nodes. They are shown to require fewer channels and exhibit lower communication, computation, and storage complexity, compared with existing approaches. Through simulation studies, we show that the proposed protocols can effectively increase throughput and reduce delay, as compared to several existing schemes, thus providing an effective solution to the low capacity problem in multi-hop wireless networks.

Keywords: Cross-layer design; Distributed channel assignment; Multi-channel medium access control; Multi-channel mobile ad hoc networks; Routing

1. Introduction

Despite recent advances in wireless technologies, today's wireless links still cannot offer the comparable data rates as their wired counterparts. The low throughput problem is further aggravated in multi-hop wireless environments due to the *intra-flow interference* introduced by adjacent transmitting nodes on the same path and *inter-flow interference* generated by transmitting nodes from neighboring paths. Two transmitting nodes within the interference range will interfere with each other. In addition, cumulative interference from a large number of transmitters outside a node's interference range will also cause low signal-to-interference-plus-noise-ratio (SINR) at the node. For instance, it has

☆ This research was supported in part by a National Science Foundation Integrated Graduate Education and Research Training (IGERT) grant (award DGE-9987586). Part of this work has been presented at the 2005 IEEE International Conference on Communications, Seoul, Korea, May 2005 [1].
* Corresponding author. Tel.: +1 334 844 1845.
E-mail addresses: michelle.gong@gmail.com (M.X. Gong), midkiff@vt.edu (S.F. Midkiff), smao@ieee.org (S. Mao).

1570-8705/$ - see front matter
doi:10.1016/j.adhoc.2007.11.011

been shown in [2] that the maximum capacity that the IEEE 802.11 MAC can achieve for a chain network could be as low as just one seventh of the nominal link bandwidth.

We observe that 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. In fact, such bandwidth aggregation has been widely used in infrastructure-based WLANs, where high-end access points are equipped with multiple interfaces that operate on different channels simultaneously [3]. In such networks, non-overlapping channels are distributed among different access points at the network planning stage [4]. However, IEEE 802.11 WLANs that operate in ad hoc mode rarely use multiple channels simultaneously. This is partly because that the IEEE 802.11 MAC is not designed to operate with multiple channels, resulting in a waste of precious network resources. As an example, an ad hoc network based on the IEEE 802.11a technology utilizes only one out of 12 available orthogonal channels, wasting more than 90% of the potentially available spectrum.

Consequently, there has been substantial interest in multi-channel MAC schemes that can achieve higher throughput by exploiting multiple available channels [3,5,6]. Some of the early works, e.g. [7,8], assume that every node has its own unique channel. Therefore, no channel assignment or selection is needed. However, in reality, the number of channels is limited and has to be carefully assigned to each node, in order to avoid contention and collisions and to enable optimal spatial reuse of available channels. Many channel assignment problems have been proven to be *NP*-complete and, thus, computationally intractable [4,9,10]. There exist only a few heuristic solutions, which have good performance under certain environments, for instance, in a static wireless network. However, these heuristic schemes suffer from inefficiency when employed in the mobile ad hoc environment [9,11].

In this paper, we present three principles for designing efficient distributed channel assignment schemes. *First*, to reduce the complexity of the channel assignment algorithm, channel assignment and routing should be jointly designed. This "cross-layer" design approach is motivated by the fact that both the channel assignment algorithm and the ad hoc routing algorithm are invoked when there is a change in the network topology. Exploring this design principle can potentially reduce the complexity of channel assignment algorithms. *Second*, channels should be assigned only to active nodes. This "on-demand" channel assignment principle is motivated by the fact that only nodes on active routes need valid channels. Some existing channel assignment schemes assign channels to *all* nodes in the network, regardless of whether they are active or not, thus requiring a large number of orthogonal channels. If this on-demand assignment principle is implemented, fewer channels (i.e. fewer resources) may be required in the network to achieve a comparable performance.

Third, the capacity of mobile ad hoc networks can be adversely affected by both "hidden terminals" and "exposed terminals." The *hidden terminal* problem occurs when transmitters outside of radio range of each other transmit at the same time, causing a collision at the receivers [10]. The *exposed terminal* problem occurs when a node is prevented from sending packets to other nodes due to a neighboring transmitter, even though this transmission will not cause interference. In addition, cumulative interference generated by nodes two or more hops away may also adversely affect channel utilization and network capacity. Thus, to improve network performance, distinct channels should be assigned such that hidden terminals, exposed terminals, and cumulative interference can be avoided as much as possible.

We present a new channel assignment protocol, named Channel Assignment Ad hoc On-demand Distance Vector routing (CA-AODV), that implements these design principles. In CA-AODV, channel assignment is combined with the AODV routing protocol and is performed in a cross-layer and on-demand fashion. Specifically, channel assignment is performed during the route discovery phase and channel information is piggybacked in the routing control messages. CA-AODV assigns different channels for neighboring nodes within a k-hop region along the same path, thus allowing concurrent transmission on neighboring links along the path and effectively reducing the intra-flow interference. We also present two extensions to the CA-AODV protocol, namely, the Enhanced 2-hop CA-AODV (E2-CA-AODV) protocol and the Enhanced k-hop CA-AODV (Ek-CA-AODV) protocol. In addition to intra-flow interference, these two extensions also aim to minimize inter-flow interference by assigning orthogonal channels to active nodes within a k-hop neighborhood, where $k \geqslant 2$. With such channel

Table 1
Notation

Symbol	Comments
$\mathscr{C}$	Set of channels
$\mathscr{A}$	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
$v_j^t \in V_t$	An active transmitter
$v_i^r \in V_r$	An active receiver
$v_{T(i)}^t$	Desired transmitter for receiver v_i^r
$P(v_j^t, v_i^r)$	Received power at v_i^r from transmitter v_j^t
$S(v_{T(i)}^t, v_j^t)$	Cross-correlation between the channels used by the two transmitters
β	SINR threshold for successful reception
P_N	Power level of additive white Gaussian noise

assignment, more concurrent transmissions are achieved for nodes along various routes within the k-hop neighborhood, thus effectively improving the throughput and delay performance. Simulation results in mobile ad hoc networks show that the performance of E2-CA-AODV approaches that of a multi-channel scheme with an unlimited number of channels (i.e., the ideal case with unlimited amount of resources). In addition, the proposed protocols exhibit lower complexity than many existing centralized and distributed approaches.

In addition to the distributed channel assignment protocols, we also develop a transmitter-based multi-channel MAC (MC-MAC) protocol that extends the benefit of channelization to multi-hop mobile ad hoc networks. Because multiple hop information or the whole network topology can be visible to the routing layer, MC-MAC can benefit from the combined routing and channel assignment scheme and offers improved network performance.

The remainder of this article is organized as follows. We formulate the channel assignment problem in Section 2. Table 1 summarizes the notation used in this paper. In Section 3, we present CA-AODV and its extensions. The MC-MAC protocol is described in Section 4. We present simulation results in Section 5, and discuss related work in Section 6. Section 7 concludes this paper.

2. Problem statement

An 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 wireless 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.

The interference range is defined to be the k-hop neighborhood of a node. Interference can be significantly reduced if nodes within the k-hop neighborhood are assigned different orthogonal channels. The k-hop neighbors of a node v is the set

$$N_k(v) = \{w \in V \mid h(v, w) \leqslant k\}, \quad (1)$$

where $h(v, w)$ is the hop distance from v to w, i.e. the minimum number of hops of any path from v to w. Note that $N_1(v)$ is the set of directly connected neighbors of node v. Let $\mathscr{C}$ denote the set of channels in the network. We further define $V_t \subset V$ to be the set of active transmitters and $V_r \subset V$ the set of active receivers. Let $v_i^r \in V_r$ be an active receiver and $v_j^t \in V_t$ be an active transmitter. If v_j^t is transmitting on the same channel on which v_i^r is receiving, but v_j^t is not the intended transmitter to v_i^r, then transmitter v_j^t will interfere with receiver v_i^r. The power associated with the interference, $P(v_j^t, v_i^r)$, is the received power at node v_i^r, which is a function of the effective transmitting power of node v_j^t, the distance between nodes v_j^t and v_i^r, and channel conditions, e.g. path loss and fading [12].

If k is set to an appropriate value and $|\mathscr{C}|$ is sufficiently large, hidden nodes and exposed nodes can be largely avoided and harmful interference can be mitigated. In this case, distributed channel assignment algorithms should assign distinct channels to any nodes within a k-hop neighborhood. However, in many cases, the number of available channels may be less than the number of nodes in a k-hop neighborhood. Therefore, two or more nodes may need to share the same channel. To balance the number of nodes sharing the same channel, the main design objective of a distributed channel assignment algorithm is to minimize the maximum number of nodes sharing the same channel with a designated node $v_j^t \in V_t$ within its k-hop neighborhood, i.e.

$$\text{Minimize} \quad \max_{v_j^t \in V_t} \{n_k(v_j^t)\} \quad (2)$$

Subject to

$$\frac{P(v_{T(i)}^t, v_i^r)}{\sum_{v_j^t \in V_t \setminus v_{T(i)}^t} [P(v_j^t, v_i^r) \cdot S(v_{T(i)}^t, v_j^t)] + P_N} \geqslant \beta, \quad (3)$$

where P_N is the power level of additive white Gaussian noise (AWGN) noise and β is the minimum SINR required for a successful packet reception. The term $S(v^t_{T(i)}, v^t_j)$ indicates the adjacent channel interference between the channels used by the desired transmitter $v^t_{T(i)}$ and transmitter v^t_j. We have $S(v^t_{T(i)}, v^t_j) = 0$ if $v^t_{T(i)}$ and v^t_j use strictly orthogonal channels; $S(v^t_{T(i)}, v^t_j) = 1$ if $v^t_{T(i)}$ and v^t_j use the same channel. Generally we have $0 < S(v^t_{T(i)}, v^t_j) < 1$.

A channel assignment protocol should distribute available channels with any pre-defined value of k in such a way that the maximum number of transmitters that share the same data channel is minimized. Meanwhile, the same set of channels should be reused in such a way that the accumulated interference generated on any particular data channel is below a certain threshold. Because the channel assignment problems are shown to be NP-complete, there exists no polynomial algorithms for general mobile ad hoc networks. Thus, efficient heuristic algorithms, including both centralized [4] and distributed [9,10], become particularly appealing. In addition, since it is very challenging to obtain global statistics in a mobile ad hoc network, a realistic channel assignment protocol should be distributed, i.e. it should operates with local information only. In the following section, we present such distributed heuristic algorithms for the formulated problem, which provide near-optimal solutions.

3. Distributed channel assignment protocols

In this section, we first describe the three design principles on distributed channel assignment and then introduce three protocols that implement these design principles. It is assumed that channel assignment is transmitter-based, meaning that distinct channels are assigned to different transmitters.[1] It has been shown that transmitter- and receiver-based channel assignment problems are essentially equivalent [9]. Therefore, the proposed design principles apply to receiver-based protocols and the proposed protocols can be easily modified to assign channels based on receivers as well.

3.1. Design principles

The first and primary principle is "cross-layer" design, where channel assignment is jointly considered with routing. This is motivated by the fact that both channel assignment and routing will be invoked when there is a topology change. In addition, piggybacking channel information in routing control messages can greatly reduce the communication overhead of channel assignment protocols. For instance, an existing channel assignment algorithm has a communication complexity of $O(d^2 \cdot |V|)$, where d is the maximum number of one-hop neighbors that a node can have and $|V|$ is the total number of nodes in the network [11]. Such a complexity implies that whenever there is a topology change, up to $O(d^2 \cdot |V|)$ messages will be exchanged in the network. Similarly, a recently proposed channel assignment protocol has a complexity of $O(K|V|^3 \log m + m^2)$ [13], where m is the total number of radio connections in the network and K is the minimum number of neighbors that a node has. Such high complexity makes it difficult to implement these protocols in a mobile ad hoc network environment where topology is constantly changing. By exploiting the cross-layer design principle, a combined Channel Assignment and AODV (CA-AODV) algorithm can reduce the communication complexity to $O(1)$, since all channel information is carried in routing control messages (see Section 3.2).

The second design principle states that channels should be assigned only to active nodes. Before a node acquires a valid route, it cannot transmit or receive data packets and, thus, does not need a channel. We call this type of node an "inactive" node, and a node in a valid route an "active" node. Most existing channel assignment schemes assign channels to *all* nodes in a wireless network, regardless whether they are active or inactive [9–11,13]. If not all nodes in a network are active at the same time, such schemes may require more wireless channels than necessary. By assigning channels *on-demand* to only active nodes, we can potentially reduce the number of channels required in a wireless network, since the number of required wireless channels is generally proportional to the number of active nodes rather than the total number of nodes in the entire network.

Finally, distinct channels should be assigned in such a way that hidden nodes, exposed nodes and interference can be avoided as much as possible. Many existing channel assignment schemes are

[1] Note that actually the channels are assigned to the links. When a transmitter has a frame to send, it will notify the corresponding receiver to tune its transceiver to the assigned channel, in order to receive the frame (see Section 4). That is, the channel is actually assigned to this link from the transmitter to the receiver.

designed to solve the hidden node problem [9–11]. However, the exposed node problem can also reduce channel utilization. Interference can also corrupt data packets and reduce throughput. We propose to assign distinct channels to any nodes within a k-hop neighborhood, where k is a design parameter that provides a suitable tradeoff between the number of required channels and the achieved interference level, and between the system performance and control overhead. Therefore, concurrent transmissions in the k-hop neighborhood can be achieved to improve the network performance.

3.2. Distributed channel assignment protocols

The proposed design principles can be applied to both reactive routing algorithms, such as Ad hoc On Demand Distance Vector (AODV) [14] protocol, and proactive routing protocols, such as the Optimal Link State Routing (OLSR) protocol [15]. In the following, we will use AODV as an example to demonstrate the three design principles. AODV is a reactive routing protocol, which means nodes do not maintain up-to-date routes to all destinations at all times. Instead, a node initiates a route discovery procedure by broadcasting a Route Request (`RREQ`) message only when it has packets for the destination node and it has no valid route to the destination. Upon receiving the RREQ message, the destination node or an intermediate node that has a valid route to the destination will send back a Route Reply (`RREP`) message to the source node. The RREP sets up a path from the source to the destination as it is forwarded back to the source node.

Throughout this paper, we assume that each node is equipped with two transceivers: one for data transmissions and the other for control messages. Control messages, such as RREQ and RREP, are sent on the control channel that is shared by all the nodes in the network. Data packets may be sent on different data channels that have been assigned to active nodes in the network. In the following, we present a combined channel assignment and AODV scheme, namely, Channel Assignment AODV (CA-AODV), and its two extensions: Enhanced 2-hop CA-AODV (E2-CA-AODV) and Enhanced k-hop CA-AODV (Ek-CA-AODV).

3.2.1. Combined channel assignment and AODV

CA-AODV, like AODV, operates in two phases: route discovery and route reply. During the route discovery phase, channel information about a node's k-hop neighbors along the same route is carried by the broadcast RREQ message. Any node that receives the RREQ message updates its next-hop table entries with respect to preceding nodes in the path back to the source. Each table entry consists of both the route and the indices of channels that have been taken, so far, by the node's k-hop neighbors on the same route. If the node has no channel assigned to it, it updates its available channel set, denoted by $\mathscr{A} \subseteq \mathscr{C}$, by marking the channels taken by the preceding k (or fewer if the route is not k hops long) nodes on the path as unavailable. Then, it randomly picks a channel from the set of available channels A. Furthermore, this node will associate a timer with these updates. If this node does not receive an RREP before the timer expires, these updates will be restored to the original states, since this node will not be included in that path.

During the route reply phase, channel information is carried in the unicast RREP message. Upon receiving the RREP packet, each node along the route updates its next-hop table entries, as well as the index of the channel to be used for this link. After the route has been established, each node along the route should have a channel that is different from any of its k-hop neighbors on the same route. As in AODV, a route expires if it is not used or reactivated for a certain period of time. The entry that corresponds to this route will then be deleted from the routing table and the channels assigned to this route will become available for re-assignment.

A flowchart that describes the operation of CA-AODV is shown in Fig. 1. Figs. 2–5 describe its four main procedures, i.e. sendRREQ(), recvRREQ(RREQ), sendRREP() and recvRREP(RREP). sendRREQ() and recvRREQ(RREQ) are invoked in the route discovery phase, while sendRREP() and recvRREP(RREP) are invoked in the route reply phase. In recvRREQ(RREQ) and recv(RREP), *getNeighborInfo()* retrieves channel information from `RREQ` or `RREP` messages received from neighboring nodes, while *randomChannel(A)* returns a channel index that is uniformly chosen from the available channel set $\mathscr{A}$. As shown in Fig. 5, a node will update its available channel set $\mathscr{A}$ upon receiving an RREP. If a channel conflict is found, the node randomly chooses a channel for itself from set $\mathscr{A}$. This node will then forward the RREP carrying its own channel and its upstream neighbors' channels.

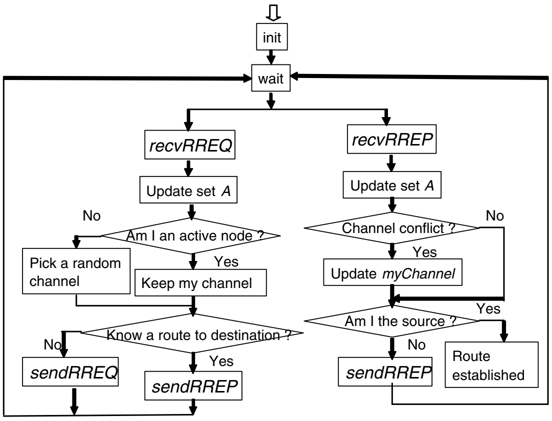


Fig. 1. Algorithm flowchart of CA-AODV.

```
Procedure sendRREQ ( )
  if (myChannel == INVALID_CHANNEL)
    myChannel = randomChannel(A);
  endif
  for (i=1 to k)
    RREQ.neighborChannelIndex[i] =
                    neighborChannel[i] ;
  endfor
  Broadcast RREQ;
endprocedure
```

Fig. 2. Procedure sendRREQ() in CA-AODV.

```
Procedure recvRREQ(RREQ)
  rt = getRouteInfo(RREQ);
  neighborChannel = getNeighborInfo(RREQ);
  if (route is not in the routing table)
    addRoute(rt);
  endif
  Update the available channel set A;
  if (myChannel == INVALID_CHANNEL)
    myChannel = randomChannel(A);
  endif
  if (myIndex == destination ||
              I know a route to destination)
    sendRREP();
  else
    sendRREQ();
  endif
endprocedure
```

Fig. 3. Procedure recvRREQ(RREQ) in CA-AODV.

CA-AODV introduces very low control overhead, since the channel information is completely piggybacked in routing control messages (e.g., as a list of channel IDs or in the form of a "bit-map"). However, CA-AODV only considers "intra-path" contention. It does not explicitly attempt to reduce

```
Procedure sendRREP( )
  if (myChannel != INVALID_CHANNEL)
    RREP.myChannelIndex = myChannel;
  else
    RREP.myChannelIndex =
                  INVALID_CHANNEL;
  endif
  for i=1 to k
    RREP.neighborChannelIndex[i] =
                  neighborChannel[i];
  endfor
  Unicast RREP;
endprocedure
```

Fig. 4. Procedure sendRREP() in CA-AODV.

```
Procedure recvRREP(RREP)
  neighborChannel = getNeighborInfo(RREP);
  Update the available channel set A;
  if (Detect a channel conflict)
    channelConflict =TRUE;
  endif
  if (channelConflict == TRUE)
    myChannel = randomChannel(A);
  endif
  RREP.myChannelIndex = myChannel;
  for i=1 to k
    RREP.neighborChannelIndex[i] =
                  neighborChannel[i];
  endfor
  Unicast RREP;
endprocedure
```

Fig. 5. Procedure recvRREP(RREP) in CA-AODV.

collisions and interference introduced by neighboring paths. It can be used when "intra-path" contention is the dominant limiting factor, e.g. in sparse networks with a few long lasting multi-hop sessions. In the rest of this section, we present two extensions of CA-AODV that can effectively handle both "intra-path" and "inter-path" contentions. We show that with a small increase in control overhead, the performance of CA-AODV can be further improved.

3.2.2. Enhanced 2-hop channel assignment and AODV (E2-CA-AODV)

To improve the performance of CA-AODV while still maintaining low control overhead, an extension of CA-AODV, i.e. E2-CA-AODV, is proposed. Unlike CA-AODV that has constant communication overhead, E2-CA-AODV has a linear communication overhead, i.e. $O(d_a)$, where d_a is the maximum number of active neighbors that a node can have. E2-CA-AODV seeks to assign distinct channels to active nodes within any *two-hop neighborhood*.

In addition to RREQ and RREP messages, E2-CA-AODV uses another AODV routing control message, the HELLO message, for channel assignment. As in AODV, HELLO messages are broadcast periodically among one-hop neighbors. If a node is active, it will indicate its assigned channel and a *NodeNumber* in its HELLO messages. Each time a node chooses a data channel, it updates its *NodeNumber* by generating a random number in $[1, M]$, where $M \gg 1$ in order to minimize the chance that two or more neighboring nodes choose the same *NodeNumber*. In addition, HELLO messages also carry the channel and *NodeNumber* information of a node's active one-hop neighbors. Upon receiving HELLO messages from one-hop neighbors, a node will learn the channel assignments within its 2-hop neighborhood, and update its available channel set $\mathscr{A}$ by removing the channels taken by active neighbors. The *NodeNumber* is used for resolving channel conflicts. If two or more active nodes choose the same channel, the node with the smallest *NodeNumber* will retain its channel while the other nodes shall randomly pick another data channel from $\mathscr{A}$. Note that *NodeNumber* is randomly generated each time a channel is chosen. So every node will have a fair chance of winning a channel when involved in a collision, while schemes using static node IDs will always favor a node with a lower (or higher) ID. Finally, the node that updates its own data channel will inform its neighbors in the next HELLO message.

The other operations of E2-CA-AODV, such as Route Request and Route Reply, are similar to those of CA-AODV. As discussed, E2-CA-AODV piggybacks channel assignment information in three types of routing control messages, i.e. RREQ, RREP, and HELLO. Therefore, E2-CA-AODV has two extra procedures, i.e. sendHello() and recvHello(HELLO), as shown in Figs. 6 and 7, for handling the Hello messages. In recvHello(HELLO), the *getActiveNeighborInfo(HELLO)* function retrieves channel information from received HELLO messages.

Because E2-CA-AODV exchanges channel information among two-hop neighbors, it can successfully mitigate both intra-flow and inter-interference with the 2-hop neighborhood, resulting in lower interference and more concurrent transmissions. We will present its performance in Section 5.

```
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
```

Fig. 6. Procedure sendHello() in E2-CA-AODV.

```
Procedure recvHello(HELLO)
  (neighborChannels, NodeNumber) =
              getActiveNeighborInfo (HELLO);
  Update the available channel set A;
  channelConflict = FALSE;
  if (Detect a channel conflict)
     if (activeNode == TRUE)
        if (activeMe == FALSE)
           channelConflict = TRUE;
        else if (NodeNumber <= myNumber)
           channelConflict = TRUE;
        endif
     else
        if (activeMe == FALSE &
              NodeNumber <= myNumber)
           channelConflict = TRUE;
        endif
     endif
  endif
  if (channelConflict == TRUE)
     myChannel = randomChannel(A);
     myNumber = randomNumber( );
  endif
endprocedure
```

Fig. 7. Procedure recvHello(HELLO) in E2-CA-AODV.

3.2.3. Enhanced *k*-hop channel assignment and AODV (E*k*-CA-AODV)

Because CA-AODV only considers collisions and interference on the same route, it does not perform well when multiple active nodes on different routes co-locate in the a k neighborhood. E2-CA-AODV seeks to both avoid collisions and mitigate intra-flow interference in the two-hop neighborhood while maintaining low control overhead. In order to further reduce *cumulative* interference from transmitters beyond the two-hop neighborhood, we propose E*k*-CA-AODV that considers both collisions and interference within a k-hop neighborhood (where $k \geqslant 2$), at the cost of higher communication overhead.

E*k*-CA-AODV introduces an extra control message called ChannelTaken. Most of the operations of Ek-CA-AODV are similar to those of E2-CA-AODV, except for the operation involving the extra control message `ChannelTaken`. If a node on an established route detects that a new route in the neighborhood is being set up, it shall broadcast a `ChannelTaken` message that carries its own channel index. The TTL of the `ChannelTaken` message is set to k to ensure that the `ChannelTaken` message be broadcast only to the current node's k hop neighbors. Upon receiving a `ChannelTaken` message, each node will update its next-hop neighbor table and the available channel set $\mathscr{A}$. If a channel conflict is detected by a node that has not yet on an established route, this node shall set a `channelConflict` flag. Once receiving a `RREP` message, a node checks to see whether the `channelConflict` flag is set. If so, the node will randomly pick another channel from the channel set $\mathscr{A}$. Through `ChannelTaken` messages, channels taken by nodes on established routes can be conveyed to other nodes in the network, up to k hop away. Therefore, conflicting channels within the k-hop neighborhood can be largely avoided, provided that the number of available channels is sufficiently large. The two additional procedures for handling `ChannelTaken` messages, i.e. sendChannelTaken() and recvChannelTaken(), are shown in Figs. 8 and 9.

To allow sufficient time for `ChannelTaken` messages to propagate to all nodes within the k-hop range, the destination node or a node that has a valid route to the destination should wait for a period of time, denoted by W_t, before returning the `RREP` message. The value of W_t provides a tradeoff between route discovery delay and the correctness of channel assignment information. We choose W_t to be a function of both k and t_p, i.e.

$$W_t = \alpha \times k \times t_p, \tag{4}$$

where $1 \leqslant \alpha \leqslant 2$ is a constant and t_p is the per hop propagation delay.

Parameter α is used to accommodate variations in t_p. Usually a large value should be used if the network is dense and traffic load is high; while small values should be used for low-load sparse networks.

Procedure ***sendChannelTaken*( )**
  ChannelTaken.channelIndex = *myChannel*;
  ChannelTaken.TTL = *k*;
  Broadcast the *ChannelTaken* message;
endprocedure

Fig. 8. Procedure sendChannelTaken() in Ek-CA-AODV.

Procedure ***recvChannelTaken*(*ChannelTaken*)**
  if (*MyChannel* == *ChannelTaken.channelIndex*)
    ChannelConflict = **TRUE**;
  endif
  Update the available channel set ***A***;
  if (*ChanelTaken.TTL*-- > 0)
    Broadcast the *ChannelTaken* message;
  endif
endprocedure

Fig. 9. Procedure recvChannelTaken (ChannelTaken) in Ek-CA-AODV.

With the assumption that a transceiver and channel is specifically used for control messages, the delay for the `ChannelTaken` message to propagate to the entire k-hop neighborhood will only be affected by the control traffic load. Given a traffic load and network density, we should choose an α value that is sufficiently large so that `ChannelTaken` messages can propagate to the entire k-hop neighborhood, while it should also be small so as to minimize the route discovery delay.

Because a `ChannelTaken` message is relayed to an active node's k-hop neighbors, in the worst case, every active node in the network may send a `ChannelTaken` message to its k-hop neighbors. Thus, the communication overhead of Ek-CA-AODV is $O(|V_t| \cdot |V_k|)$, where $|V_t|$ is the number of active nodes in the network and $|V_k|$ is the number of nodes in a k-hop neighborhood. Both the computation overhead and the storage overhead of Ek-CA-AODV are $O(|V_k|)$ since each node only needs to process and store channel information of its k-hop neighbors, denoted by $|V_k|$. Due to its higher communication overhead, Ek-CA-AODV is best suited for a mobile ad hoc network with low mobility, where link breakage is not so often and channels do not need to be frequently re-assigned to nodes. However, Ek-CA-AODV has its strength in the case when interference range is more than 2-hops and when cumulative interference needs to be considered, as compared to the previous two schemes.

3.3. Protocol analysis

In this section, we first derive an upper bound on the number of distinct channels required by Ek-CA-AODV ($k \geqslant 2$). We then prove the correctness of Ek-CA-AODV. The proof for CA-AODV is similar and is omitted for brevity.

Proposition 1. *To assign distinct channels to any node within a k-hop range, the number of channels required has an upper bound of* $n_r \cdot (k+1)$, *where* n_r

is the number of active routes that lie within the k-hop range of any node in the network.

Proof. This proposition can be proven by induction.

(i) Base Case: If there is only one route in the network, it can be easily shown that the number of required distinct channels is $k + 1$.
(ii) Induction Step: Assume that when there are n active routes within a k-hop range, the required number of channels is $n \cdot (k + 1)$. If there are $n + 1$ active routes within a k-hop range, the $(n + 1)$th route can be assigned $k + 1$ new channels that are different from any of the previous $n \cdot (k + 1)$ channels. The total number of channels needed for $n + 1$ active routes is then $(n + 1) \cdot (k + 1)$.

Therefore, the proposed algorithms need at most $n_r \cdot (k + 1)$ distinct channels. □

Note that this upper bound is achieved when there is no common links among active routes. If that is the case, channels are assigned to each active route independently and each route requires $k + 1$ channels. If active routes in the network share links, the network requires fewer channels than $n \cdot (k + 1)$.

Proposition 2. *After a new route has been established, each node along the new route is assigned a distinct channel among its k-hop neighbors, provided that both the number of available channels and* W_t *is sufficiently large.*

Proof. Under the assumption that the channel assignment procedure is not disrupted by sudden failure or malfunction of nodes, the channel information carried by control messages is consistent with the channel information saved at each node. Moreover, `ChannelTaken` messages can propagate to all nodes within a k-hop range, provided that W_t is sufficiently large. When a new route is established, each node on the route will have its k-hop active neighbors' channel information. Because a node randomly picks its own channel from the available channel set, which does not contain any of its k-hop neighbors' channels, this node must have a channel that is distinct from any of its k-hop neighbors. □

Many existing algorithms do not explicitly consider the case when the number of available channels is not sufficiently large [9–11]. The proposed algorithms can effectively handle this case. A general rule would be to let the nodes using the same channel to be as far apart from each other as possible. Specifically, if $k \leqslant 2$, a node with $\mathscr{A} = \emptyset$ should randomly pick a channel from the set of least reused channels (in its k-hop neighborhood) in $\mathscr{C}$. If $k \geqslant 3$, this node should randomly pick a channel from the channels that are taken by nodes two-hop away. Thus, the algorithms ensure minimum number of collisions, as well as taking interference into consideration (see Eqs. (2) and (3)).

4. Description of the MC-MAC protocol

The proposed channel assignment and routing protocols have a companion multi-channel MAC (MC-MAC) protocol, which is designed to allow simultaneous data transmissions on different data channels. It is assumed that there is one dedicated control channel and up to N data channels in the network. Each data channel is equivalent and has the same bandwidth. Recall that each host is equipped with two transceivers, Transceiver I operating on the common control channel all the time and Transceiver II that switches from one channel to another for data packets.

MC-MAC is a transmitter-based protocol. As discussed, nodes are assigned channels by the combined channel assignment and routing protocols. RTS and CTS messages are sent on the control channel with Transceiver I, while data packets and ACKs are sent on the assigned data channel by Transceiver II. When a node is ready to transmit, it will first convey its assigned data channel to the destination node through RTS/CTS exchange. As shown in Fig. 10, when a sender, say node A, intends to transmit, its Transceiver I will broadcast an RTS message carrying its own data channel index, c_{AB}. Upon receiving the RTS message, the

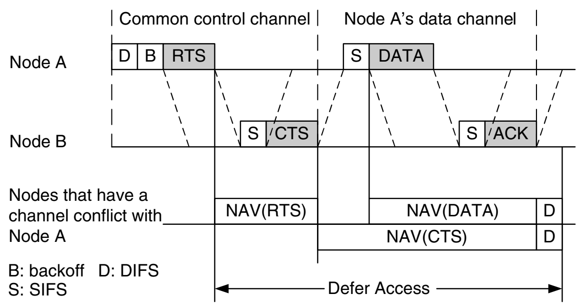


Fig. 10. The four-way handshake procedure of MC-MAC.

destination, say Node B, replies a CTS message carrying c_{AB} from its Transceiver I and sets its Transceiver II to channel c_{AB}. After Node A receives the CTS from the control channel, its Transceiver II switches to the confirmed data channel c_{AB} 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 RTS/CTS exchange. If a node is assigned the same data channel c_{AB}, two situations may happen, as illustrated in Fig. 10.

– If the node overhears a CTS message, it should defer from using the data channel c_{AB} 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 the data channel c_{AB}. 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 this channel immediately.

The sender listens on the data channel until an ACK is received or a timeout occurs. If a node receives an RTS on the control transceiver while its data transceiver is busy communicating with another node, it returns a Negative CTS (NCTS) on the control channel to the sender, indicating that a collision has not occurred. Thus the sender is not obliged to increase its contention window nor to back off.

Because MC-MAC transmits control packets on the common control channel and data packets on non-overlapping orthogonal data channels, the transmissions on the control channel and on different data channels can occur in parallel. This type of parallelism is sometimes called “pipelining” [16]. When nodes have different data channels from the on-going data transmissions, they need to defer only for the duration of control packet transmission. Since the size of data packets is usually much larger than that of control packets, many data transmissions can occur in parallel on different data channels. Also note that MC-MAC solves the hidden terminal problem as in the IEEE 802.11 MAC by using the RTS/CTS dialog to reserve data channels.

5. Performance evaluation

In this section, we first compare the performance of E2-CA-AODV with that of an existing channel assignment protocol, i.e. the channel assignment scheme (CAS) [11]. We then demonstrate the capacity improvement of E2-CA-AODV and MC-MAC over the original IEEE 802.11 MAC. All the simulations reported in this section are performed using the ns-2 simulator [17,18].

5.1. Simulation setting

We assume that 64 wireless nodes are placed randomly in a square area. Each node is equipped with two half-duplex transceivers. All nodes in the network share the same common control channel. There are 6 or 12 different data channels available. In all simulations, the radio range of a node is set to 250 m and the interference range is set to 550 m, which is approximately twice the radio range [18]. The two-ray ground propagation model is selected and the physical channel bandwidth for all data channels and the control channel is set to 2 MB/s [18]. Most current wireless LAN cards have a channel switch delay of 40–80 μs [19], and we therefore assume a channel switch delay of 80 μs. Each simulation lasts for 7600 s, where the warm-up period is 3600 s and the effective simulation time is 4000 s, in order to get steady state statistics [20].

Four UDP flows are generated in the network. Each UDP flow has an offered load ranging from 40 KB/s to 1000 KB/s. For routing, AODV is used with the IEEE 802.11 MAC and CAS, while E2-CA-AODV is used with MC-MAC. We choose CAS because it is one of a few distributed channel assignment protocols that can operate in the mobile, multi-hop environment. Additionally, unlike other channel assignment schemes that were proposed to work with the IEEE 802.11 MAC [3,13,21], CAS can work with the proposed MC-MAC.

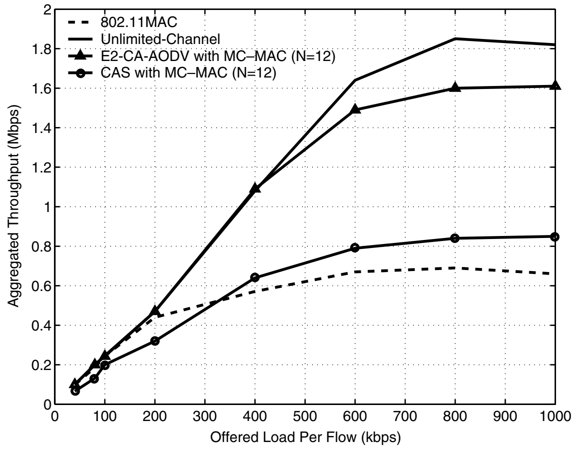
Two wireless ad hoc networks are simulated: a 800 m × 800 m dense network and an 1600 m by 1600 m sparse network. Mobile nodes move randomly according to the random waypoint mobility model, where the maximum speed is 5 m/s and the minimum speed is 4 m/s. The maximum pause time is 5 s. Although we have also studied E2-CA-AODV and CAS at higher speeds, the results are not significantly different. We find that the network density, on the other hand, has a much greater impact on the performance than node speed. This coincides with observations made by Bahl, Chandra, and Dunagan in [19].

5.2. Performance comparison

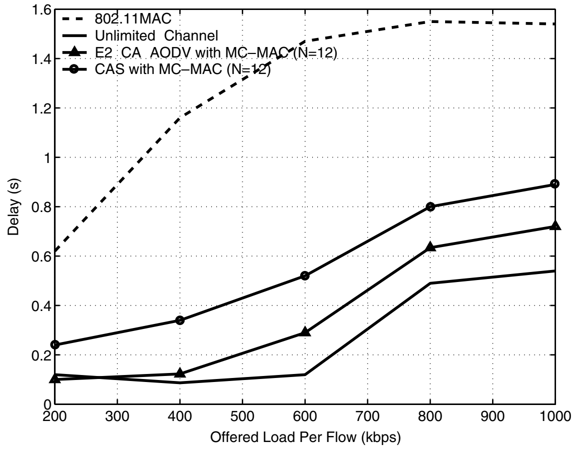
We first compare E2-CA-AODV with three schemes: (i) the single-channel IEEE 802.11 MAC, (ii) a scheme with unlimited number of data channels, and (iii) CAS. The IEEE 802.11 MAC serves as a *lower bound* for our performance study. In the unlimited data channel scheme, each node has its own unique data channel in the network and a common control channel is shared by all nodes in the network. This is the ideal case with unlimited network resource and serves as a performance *upper bound*: its performance upper bounds that of any distributed/centralized channel assignment protocols with a finite set of channels.

CAS assigns distinct channels, or codes, to a node and its *two-hop* neighbors. In CAS, each node sends out code assignment messages (CAM) that propagate to its one-hop neighbors. CAMs are transmitted under three conditions: (i) when a new node comes up, (ii) when a node detects a change of code by any of its one-hop neighbors, and (iii) when a node finds that one of its one-hop neighbors is no longer active. A CAM contains: (i) the address and code of the source node, (ii) the addresses and codes of source node's one-hop neighbors, (iii) the acknowledgements to earlier received CAMs, and (iv) 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 [11]. In contrast, E2-CA-AODV has a linear complexity $O(d_a)$, since channel information is piggybacked in routing control messages.

Fig. 11 plots the performance of the above schemes in the dense network, and Fig. 12 plots the performance of the schemes in the sparse network. For aggregate throughput, Figs. 11a and 12a show that because of its high control overhead, CAS performs worse than E2-CA-AODV in both dense and sparse networks. Especially in a dense network where a node may have many neighbors, the control overhead of CAS is so high that when the data rates are lower than 300 Kbps, CAS with 12 data channels performs even worse than the single-channel IEEE 802.11 MAC (see Fig. 11a). That is, the performance degradation caused by the control overhead dominates the performance gain of using multiple channels. Similar observation can be made in Fig. 12a for rates lower than 200 Kbps.



(a) Aggregate throughput

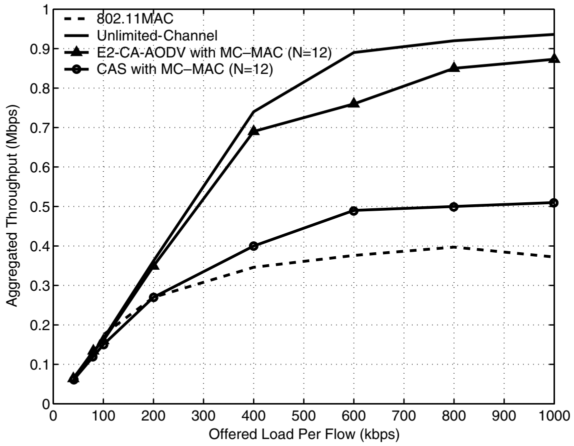


(b) Average end-to-end packet delay

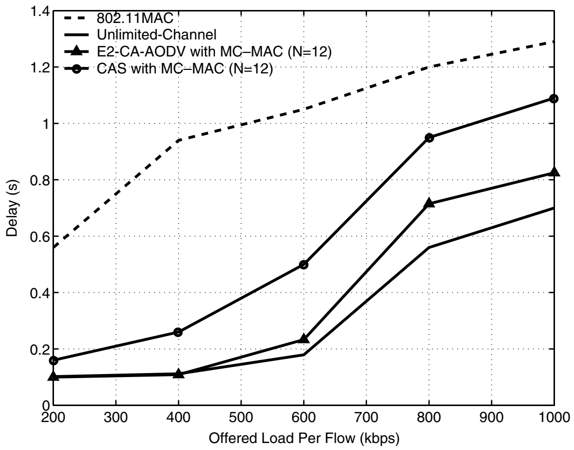
Fig. 11. E2-CA-AODV vs CAS in a dense network. (a) Aggregate throughput and (b) average end-to-end packet delay.

Figs. 11b and 12b show that the delay performance of CAS is better than that of the IEEE 802.11 MAC protocol, but still not as good as that of E2-CA-AODV. Using multiple channels, both E2-CA-AODV and CAS can effectively reduce collisions and contention in the network. Thus, the end-to-end delay suffered by a data packet is greatly reduced, as compared with the single-channel IEEE 802.11 MAC.

We then compare E2-CA-AODV with a random-AODV algorithm. In the random-AODV algorithm, a node determines its own channel index based on its MAC address. Therefore, no channel assignment is needed in the random-AODV scheme, but there will be more collision and severer interference

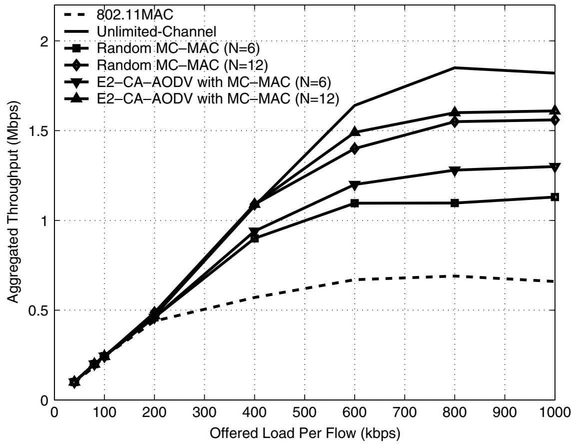


(a) Aggregate throughput

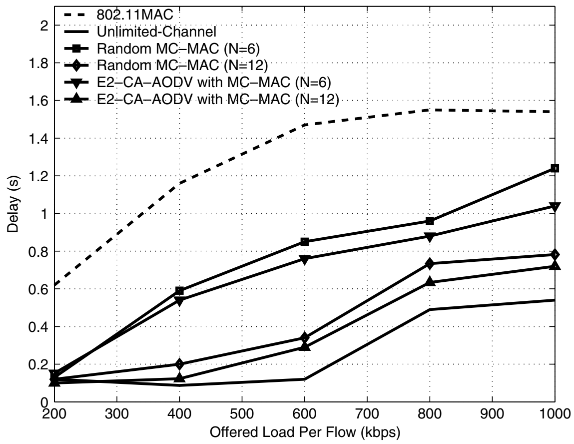


(b) Average end-to-end packet delay

Fig. 12. E2-CA-AODV vs CAS in a sparse network. (a) Aggregate throughput and (b) average end-to-end packet delay.



(a) Aggregate throughput



(b) Average end-to-end packet delay

Fig. 13. Performance of E2-CA-AODV in a dense network. (a) Aggregate throughput and (b) average end-to-end packet delay.

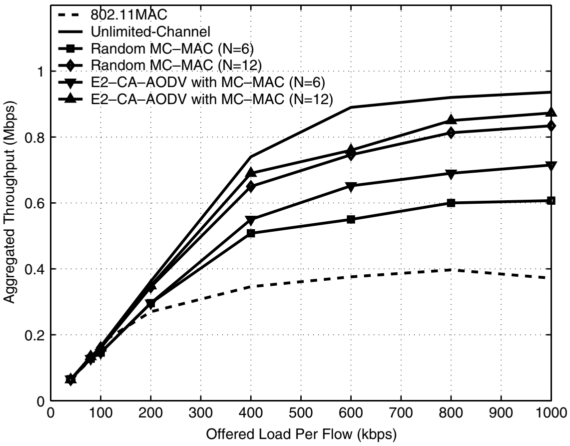
since the channels are randomly choosing without any coordination.

The simulation results for the dense network and the sparse network are presented in Figs. 13 and 14, respectively. Fig. 13a shows that E2-CA-AODV always achieves a higher throughput than the random-AODV scheme in a dense network, given an equal number of channels. The performance gap between the two is larger as the number of available channels decreases. We also find that E2-CA-AODV combined with MC-MAC can have a throughput up to three times higher than that of IEEE 802.11 MAC. Fig. 13b shows that the end-to-end delay increases for all schemes when the data rate increases. However, both multi-channel schemes with 12 data channels have delays much lower than that of IEEE 802.11 MAC.

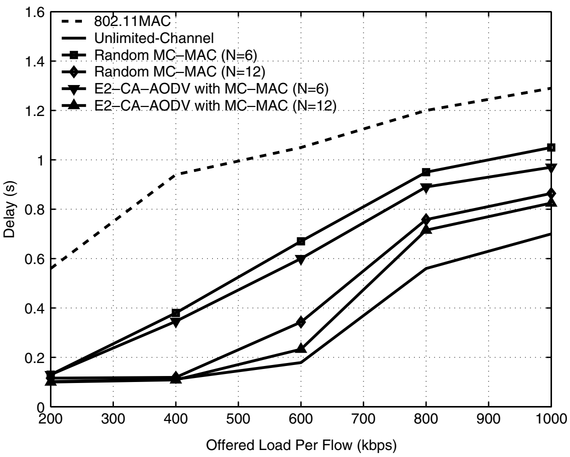
Fig. 14a and b shows the throughput and delay for all schemes in a sparse network. The performance gaps between different schemes are not as large as those in the dense network, although E2-CA-AODV still achieves an clear improvement over the random-AODV scheme. In a sparse network, the interference and collisions generated by neighboring nodes is much lower compared to those in a dense network. This is because each node has fewer neighbors in a sparse network. Therefore, the performance gain of the multi-channel MAC schemes is not as significant as in a dense network.

5.3. Remarks

Based on the simulation results presented in the previous section, we can make several interesting



(a) Aggregate throughput



(b) Average end-to-end packet delay

Fig. 14. Performance of E2-CA-AODV in a sparse network. (a) Aggregate throughput and (b) average end-to-end packet delay.

observations, which are discussed in the following. *First*, communication overhead has a profound impact on the performance of distributed channel assignment protocols. Both E2-CA-AODV and CAS seek to assign distinctive channels to nodes in a two-hop neighborhood. However, because E2-CA-AODV has lower communication overhead, it has much better performance than CAS in terms of throughput and delay, and the performance gain is more significant in a dense network. On the other hand, 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 the use of multiple channels increases the possibility of concurrent transmissions in the network.

Second, in all simulated scenarios, the performance gap between the scheme with an unlimited number of channels and E2-CA-AODV with 12 data channels, is not significant. Thus, we can conclude that due to the negligible interference generated by distant nodes, a large number of data channels is not necessary to achieve the most benefits of the use of multiple channels. The optimal number of channels should be a function of node density and interference range.

Third, the performance gap between E2-CA-AODV and the random scheme decreases when the number of available channels increases. By intelligently assigning channels, E2-CA-AODV can effectively avoid collisions and mitigate intra-flow interference. The random scheme merely tries to make use of all the available channels without coordination among the nodes and consideration of collision and interference. However, when there is a large number of available channels, collisions may be infrequent even if channels are assigned randomly.

Fourth, the performance gains achieved by of E2-CA-AODV/MC-MAC over AODV/IEEE 802.11 MAC, is not in proportion to the number of channels utilized. For instance, for E2-CA-AODV/MC-MAC with 12 available data channels, the throughput gain can be up to three times, rather than 12 times, as compared to the throughput achieved by the AODV/IEEE 802.11 MAC. MC-MAC assumes that all nodes in the network share a common control channel. With an increase in the number of data channels, the control traffic also increases and cause congestion and collisions in the common control channel, as more nodes try to transmit in parallel. Therefore, the time period a node can spend on transmitting data packets is reduced, which degrades the performance of E2-CA-AODV. The control channel becomes the performance bottleneck. In addition, the channel switch delay is non-negligible, which further degrades the performance. In fact, Kyasanur and Vaidya [22] show that when the number of transceivers is less than the number of available channels, network capacity may even be worse.

6. Related work

The channel assignment problem has been widely studied in the context of infrastructure-based wireless networks, such as cellular networks [23] and IEEE 802.11 WLANs [4]. This class of work considers the interference among access points or base stations, and focuses on assigning channels to the base stations to reduce interference and accommodate a given network traffic load. Therefore, such schemes belong to the network planning paradigm, which is quite different from the dynamic, infrastructureless ad hoc network environment considered in this paper.

In [9], Hu studies the problem of distributed code assignment for CDMA packet radio networks, including ad hoc networks. Under the assumption that each node has a neighbor table updated by a network-layer routing protocol, Hu's approach transforms the code assignment problem into a graph theory problem. This problem is shown to be *NP*-complete and fast heuristic algorithms are developed. Even though the solutions proposed are sound, they have high time complexity and high communication overhead. The schemes do not consider the case when the number of codes is limited and perfect assignment is not possible. Moreover, Hu considers only static networks in his designs.

Garcia-Luna-Aceves and Raju [11] describe a distributed code assignment scheme (CAS) that works in a mobile ad hoc network (see the previous section). CAS assigns distinct channels, or codes, to a node and its two-hop neighbors. If the number of codes available for assignment is at least $d(d-1)+2$, where d is the maximum degree, i.e., the maximum number of neighbors for any node, it is shown that there will be no interference after the algorithm converges. However, this algorithm incurs high communication and computation overhead, as well as high time and storage complexity. This algorithm does not explicitly 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 [3,13,21,24]. 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 focus of these approaches is routing, the performance of the channel assignment schemes has not been studied [3,21,24]. In addition, some channel assignment protocols have high time complexity, e.g. $O(K|V|^3 \log m + m^2)$ [13], where m is the total number of radio connections in the network and K is the minimum number of neighbors that a node has.

Table 2 compares several heuristic algorithms in terms of their communication, computation, and storage complexity. In the table, d is the maximum number of one-hop neighbors for any node (i.e. degree), $d_a \leqslant d$ is the maximum number of active one-hop neighbors for any node, k is a constant that represents a neighborhood size, $|V_t|$ is the number of active nodes in the network, $|V_k|$ is the number of nodes in a k-hop neighborhood, and n is the total number of active routes within a k-hop range of each other. For CA-AODV and E2-CA-AODV, the channel assignment information is piggybacked

Table 2
Comparison of CA-AODV to existing algorithms

Protocols	No. Channels	Commun. complexity	Comput. complexity	Storage complexity
Centralized greedy algorithm [9]	$d(d-1)+1$	N/A	$d^2 \cdot \|V\|$	$d^2 \cdot \|V\|$
Distributed channel assignment [11]	$d(d-1)+2$	$d^2 \cdot \|V\|$	d^2	d^2
Random scheme	N/A	$O(1)$	$O(1)$	$O(1)$
CA-AODV	$k+1$	$O(k)$	$O(k)$	$O(k)$
E2-CA-AODV	$d_a(d_a-1)+1$	$O(d_a)$	$O(d_a^2)$	$O(d_a)$
Ek-CA-AODV	$n(k+1)$	$O(\|V_t\| \cdot \|V_k\|)$	$O(\|V_k\|)$	$O(\|V_k\|)$

Table 3
Comparison of MC-MAC to existing multi-channel MAC protocols

Protocols	Medium access	Channel selection	Hardware requirement	Synchro. required
DCA [5]	CSMA/CA	Per packet	2 transceivers	No
MMAC [25]	CSMA/CA	Per beacon interval	1 transceiver	Yes
Multi-channel CSMA [6]	CSMA/CA	Per packet	1 transmitter multiple receivers	No
RICH-DP [26]	Channel hopping	Hopping sequence	1 transceiver	Yes
SSCH [19]	Channel hopping	Hopping sequence	1 transceiver	Yes
MC-MAC	CSMA/CA	Per route change	2 transceivers	No

in routing messages. In addition, channel assignments for each neighbor node are saved along with each entry in the neighbor table. Therefore, there is very little communication overhead and storage overhead for this two protocols. Ek-CA-AODV introduces a `ChannelTaken` message that is broadcast by active nodes to nodes within a k-hop neighborhood. Thus, the communication complexity of Ek-CA-AODV is $O(|V_t| \cdot |V_k|)$.

A significant body of prior work examines the benefits of utilizing multiple channels [3,5,6,19,25,26]. These existing multi-channel MAC protocols differ in their (i) channel selection techniques, (ii) medium access control schemes, and (iii) hardware requirements [27]. Table 3 summarizes several important features of several existing multi-channel MAC protocols. There is no general rule as to which scheme is better than another. However, a fundamental tradeoff exists between the hardware complexity and the system performance. Simpler schemes with less hardware requirements are often compatible with the IEEE 802.11 standard and are easy to implement, whereas complex schemes with greater hardware requirements often yield better performance. Because we combine channel assignment with routing protocols, the MC-MAC protocol only needs to implement the medium access control function, which greatly simplifies the protocol design.

7. Conclusions

We examined distributed, on-demand routing and channel assignment in multi-channel mobile ad hoc networks. Three principles have been presented for designing effective channel assignment protocols. Based on these design principles, we introduced CA-AODV, and its two extensions, E2-CA-AODV and Ek-CA-AODV, that combine channel assignment with on-demand routing. These protocols exhibit lower communication, computation, and storage complexity, and require fewer channels than many existing channel assignment algorithms. We also introduced a companion MC-MAC protocol that works with the proposed channel assignment protocols. Simulation results show that the proposed schemes can offer an improvement up to a factor of three in throughput over the IEEE 802.11 MAC protocol. The proposed approach provides an effective solution to the low throughput problem in multi-hop wireless networks.

References

[1] M. Gong, S. Midkiff, S. Mao, Design principles for distributed channel assignment in wireless ad hoc networks, in: Proceedings of the 2005 IEEE International Conference on Communications (ICC), Seoul, Korea, 2005, pp. 3401–3406.

[2] J. Li, C. Blake, D.D. Coute, H. Lee, R. Morris, Capacity of ad hoc wireless networks, in: Proceedings of the IEEE/ACM International Conference on Mobile Computing and Networking (MobiCom), Rome, Italy, 2001, pp. 61–69.

[3] A. Raniwala, T. Chiueh, Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network, in: Proceedings of the IEEE INFOCOM'05, Miami, FL, 2005, pp. 2223–2234.

[4] K. Leung, B.-J. Kim, Frequency assignment for IEEE 802.11 wireless networks, in: Proceedings of the IEEE VTC'03, Orlando, FL, 2003, pp. 1422–1426.

[5] S.-L. Wu, C.-Y. Lin, Y.-C. Tseng, J.-P. Sheu, 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), Dallas/Richardson, TX, 2000, pp. 232–237.

[6] N. Jain, S.R. Das, A. Nasipuri, A multichannel CSMA MAC protocol with receiver-based channel selection for multihop wireless networks, in: Proceedings of the IEEE International Conference on Computer Communications and Networks (IC3N), Dallas, TX, 2001, pp. 432–439.

[7] E. Sousa, J. Silvester, Spreading code protocols for distributed spread-spectrum packet radio networks, IEEE Transactions on Communications 36 (3) (1988) 272–281.

[8] M. Joa-Ng, I.-T. Lu, Spread spectrum medium access protocol with collision avoidance in mobile ad hoc wireless network, in: Proceedings of the IEEE INFOCOM'99, New York, NY, 1999, pp. 21–25.

[9] L. Hu, Distributed code assignments for CDMA packet radio networks, IEEE/ACM Transactions on Networking 1 (6) (1993) 668–677.

[10] A. Bertossi, M. Bonuccelli, Code assignments for hidden terminal interference avoidance in multihop packet radio networks, IEEE/ACM Transactions on Networking 3 (4) (1995) 441–449.

[11] J. Garcia-Luna-Aceves, J. Raju, Distributed assignment of codes for multihop packet-radio networks, in: Proceedings of the IEEE MILCOM'97, Monterey, CA, 1997, pp. 450–454.

[12] T. Rappaport, Wireless Communications: Principles and Practice, second ed., Prentice Hall, Upper Saddle River, NJ, 2002.

[13] J. Tang, G. Xue, W. Zhang, Interference-aware topology control and QoS routing in multi-channel wireless mesh networks, in: Proceedings of the ACM MobiHoc'05, Urbana-Champaign, IL, 2005, pp. 68–77.

[14] C. Perkins, E. Belding-Royer, S. Das, Ad hoc On-Demand Distance Vector (AODV) routing, IETF RFC 3661, July 2003.

[15] T. Clausen, P. Jacquet, Optimized Link State Routing Protocol, IETF RFC 3626, October 2003.

[16] X. Yang, N. Vaidya, A wireless MAC protocol using implicit pipelining, IEEE Transactions on Mobile Computing 5 (3) (2006) 258–273.

[17] VINT Group, UCB/LBNL/VINT network simulator ns. Available online: <http://www.isi.edu/nsnam/ns/>.

[18] The Rice University Monarch Project, Wireless and mobility extension to ns. Available online: <http://www.monarch.cs.rice.edu/cmu-ns.html>.
[19] P. Bahl, R. Chandra, J. Dunagan, SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad hoc wireless networks, in: Proceedings of the ACM MobiCom'04, Philadelphia, PA, 2004, pp. 216–230.
[20] J. Yoon, M. Liu, B. Noble, Random waypoint considered harmful, in: Proceedings of the IEEE INFOCOM'03, San Francisco, CA, 2003, pp. 1312–1321.
[21] P. Kyasanur, N. Vaidya, Routing and interface assignment in multi-channel multi-interface wireless networks, in: Proceedings of the IEEE WCNC'05, New Orleans, LA, 2005, pp. 2051–2056.
[22] P. Kyasanur, N. Vaidya, Capacity of multi-channel wireless networks: impact of number of channels and interfaces, Technical Report, University of Illinois at Urbana-Champaign, 2005.
[23] I. Katzela, M. Naghsineh, Channel assignment schemes for cellular mobile telecommunication systems: a comprehensive survey, IEEE Personal Communications 3 (3) (1996) 10–31.
[24] J. So, N. Vaidya, Routing and channel assignment in multi-channel multi-hop wireless networks with single-NIC devices, Technical Report, University of Illinois at Urbana-Champaign, 2005.
[25] J. So, N. Vaidya, Multi-channel MAC for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver, in: Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), 2004, pp. 222–233.
[26] A. Tzamaloukas, J. Garcia-Luna-Aceves, A receiver-initiated collision-avoidance protocol for multi-channel networks, in: Proceedings of the IEEE INFOCOM'01, Anchorage, AK, 2001, pp. 189–198.
[27] M. Gong, S. Mao, S. Midkiff, B. Hart, Medium access control in wireless mesh networks, in: Y. Zhang, J. Luo, H. Hu (Eds.), Wireless Mesh Networking: Architectures, Protocols and Standards, Auerbach Publications, New York, NY, 2006, pp. 147–182 (chapter 5).

Michelle Xiaohong Gong received her Ph.D. from Virginia Tech in 2005, her M.S. from University of Hawaii in 2000, and her B.S. from Wuhan University in 1996, all in Electrical Engineering. From 2005 to 2007, she worked as a system architect in the CTO group of the Wireless Networking Business Unit, Cisco Systems. Currently, she is a research scientist in Networking Technology Lab, Intel Corporation. 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 eight 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. He is now with the National Science Foundation (NSF) as a Program Director for the Integrative, Hybrid and Complex Systems (IHCS) Program of the Electrical, Communications and Cyber Systems (ECCS) Division in the Directorate for Engineering (ENG).

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.

His research interests include cross-layer design and optimization in multi-hop wireless networks, cognitive networks, and multimedia communications. He is on the Editorial Board of the Hindawi Advances in Multimedia Journal and the Wiley International Journal of Communication Systems. 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).