Design Principles for Distributed Channel Assignment in Wireless Ad Hoc Networks

Michelle X. Gong Scott F. Midkiff Shiwen Mao
The Bradley Department of Electrical and Computer Engineering
Virginia Polytechnic Institute and State University, Blacksburg, VA 24061 USA
{mgong, midkiff, smao}@vt.edu

Abstract— **Although it has been an active research area for a number of years, distributed channel assignment remains a challenging problem and existing protocols tend to be complex and usually not suitable for practical implementation. In this paper, we propose three principles that facilitate the design of efficient distributed channel assignment protocols in wireless ad hoc networks. Protocols that implement these design principles are shown to require fewer channels and exhibit significantly lower communication, computation, and storage complexity, compared with existing approaches. As examples, we present two such protocols built on the Ad-hoc On-demand Distance Vector (AODV) routing protocol. In addition, we prove the correctness of the algorithms and derive an upper bound on the number of channels required to both resolve collisions and mitigate interference. Simulation results show that, in many cases, the performance of the proposed protocols can approach that of centralized near-optimal algorithms while maintaining low control overhead.**

I. INTRODUCTION

One of the challenges in wireless ad hoc networking research arises from the distributed nature of such networks. Because no central control node is available to assign radio resources, mobile nodes need to contend to transmit. Contention-based medium access control (MAC) protocols perform poorly in a multiple-hop wireless environment due to collisions and interference caused by multi-hop routing. Recent studies have shown that MAC protocols based on carrier sense multiple access (CSMA), such as IEEE 802.11's MAC protocol, suffer low channel utilization in a multi-hop wireless environment [1]. Collisions and interference can be largely avoided by means of multiple-channel medium access schemes, where channels are carefully assigned to neighboring nodes to achieve efficient spatial channel reuse and to avoid collisions.

The need for spatial reuse of available channels motivates research on distributed channel assignment. However, despite being a subject of many years of research, distributed channel assignment remains a challenging problem. Many distributed channel assignment problems are proven to be $\mathcal{NP}$-complete and, thus, computationally intractable [2], [3]. There exist only a few heuristic solutions, none of which is efficient, especially when employed in the mobile ad hoc environment [2]–[4]. Some solutions have such high complexity that even simulation studies become difficult [4]. In addition, existing algorithms assume that the number of channels is sufficiently large. Since they do not consider the case when there are not enough channels, convergence may be a problem.

Furthermore, most of distributed channel assignment protocols only consider secondary collisions, since they are mainly designed to solve the "hidden terminal" problem [2]. Nevertheless, primary collisions and interference are also important factors that adversely affect channel utilization and network capacity. Failing to take primary collisions and interference into consideration, existing channel assignment algorithms may suffer performance degradation, especially in a multi-hop environment. Primary collisions occur when two neighboring nodes transmit to each other at the same time, whereas secondary collisions occur when transmitters outside of radio range of each other, also called "hidden terminals," transmit to the same receiver [2]. Primary collisions can be significantly reduced by using random access protocols, such as ALOHA or CSMA, while secondary collisions can be avoided by multi-channel medium access schemes or handshake mechanisms, such as request-to-send/clear-to-send (RTS/CTS). However, interference generated by nodes that are two or more hops away cannot be avoided by random access schemes or handshake mechanisms and, thus, such interference is potentially more harmful.

In this paper, we propose 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 designed together. This "cross-layer" design approach is motivated by the fact that both the channel assignment algorithm and the ad hoc routing algorithm must be invoked when there is a change in the network topology. Exploring this design principle can greatly reduce the complexity of channel assignment algorithms.

Secondly, 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. Fewer channels may be required in the network if this "on-demand" assignment principle is implemented.

Finally, both primary and secondary collisions and interference should be taken into consideration. To mitigate interference as well as to resolve collisions, distinct channels should be assigned in a way that collisions and interference can be avoided as much as possible.

As examples, we present two protocols that combine channel assignment with the AODV routing protocol, which im-

0-7803-8938-7/05/$20.00 (C) 2005 IEEE

plement these design principles. We previously described CA-AODV in [5] and, in this paper, propose a new enhanced version called ECA-AODV. Simulation results reveal that, in some cases, the performance of the combined routing and channel assignment protocols approaches that of centralized near-optimal algorithms. In addition, these two protocols exhibit much lower control overhead and complexity than both centralized approaches and other existing distributed approaches.

The remainder of this paper is organized as follows. We formulate the channel assignment problem and define performance metrics in Section II. In Section III, we describe the three design principles in detail and present two example algorithms. We analyze the performance of the proposed protocols in Section IV, and present simulation results in Section VI. Section VII is an overview of related work. Finally, we draw conclusions in Section VIII.

II. PROBLEM FORMULATION AND PERFORMANCE METRICS

A. Network Model and Problem Description

A wireless ad hoc network can be modeled as an undirected 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.

Two or more wireless nodes may generate primary collisions if they are one hop away, while secondary collisions can be generated by nodes that are two hops away. Such collisions can be eliminated if active nodes within two-hop range of each other transmit on orthogonal channels. If we define the interference range to be the k-hop neighborhood of a node, interference can be significantly reduced if nodes within the k-hop neighborhood are assigned orthogonal channels. Parameter k is a user-defined neighborhood size. There is an optimal range for k. A large value of k beyond the optimal range will not improve system performance due to the negligible interference generated by distant neighbors.

In the following, we define two performance metrics, f_{CA} and R_{CA}, as performance measures of distributed channel assignment protocols. First, we define the k-hop neighbors of a node v to be the set $N_k(v) = \{w \in V | l(v, w) \leq k\}$, where $l(v, w)$ is the distance from v to w, i.e. the minimum length of any path from v to w. Note that $N_1(v)$ is the set of directly connected neighbors of node v. $\mathcal{C}$ is defined to be the set of all available channels in the network.

B. Performance Metrics

Given a fixed number of channels, $|\mathcal{C}|$, one goal of channel assignment algorithms is to minimize the average number of nodes sharing the same channel with the designated node among this node's k-hop neighbors:

$$f_{CA} = \frac{1}{|V|} \sum_{v \in V} n_k(v). \tag{1}$$

Here, $n_k(v)$ is the number of nodes in $N_k(v)$ that share the same channel with node v.

If k is set to an appropriate value and $|\mathcal{C}|$ is large enough, both primary and secondary collisions can be largely avoided and harmful interference can be mitigated. Therefore, the channel assignment problem can be formulated from the perspective of reducing interference, which is, given $|\mathcal{C}|$, minimizing the average accumulated interference at each receiver. Here, we assume that the channel assignment scheme is transmitter based, i.e., channels are assigned to transmitters and receivers must "tune" to the channel assigned to the transmitter.

Let $V_t \subset V$ be the set of active transmitters and $V_r \subset V$ be the set of active receivers. Note that a node cannot be a transmitter and a receiver at the same time, so membership in V_t and V_r is mutually exclusive. Let $v_{r,i} \in V_r$ be a particular receiver and $v_{t,j} \in V_t$ be a particular transmitter. If $v_{t,j}$ is transmitting on the same channel on which $v_{r,i}$ is receiving, but $v_{t,j}$ is not the intended transmitter to $v_{r,i}$, then transmitter $v_{t,j}$ will interfere with receiver $v_{r,i}$. The power associated with the interference, $P(v_{t,j}, v_{r,i})$, is the received power at node $v_{r,i}$, which is a function of the distance between nodes $v_{t,j}$ and $v_{r,i}$, and is determined by the path loss model. The average interference at active receivers is determined as follows.

$$R_{CA} = \frac{1}{|V_r|} \sum_{v_{r,i} \in V_r} \sum_{v_{t,j} \in V_t \setminus v_{t,T(i)}} \left[P(v_{t,j}, v_{r,i}) \cdot S(v_{t,j}, v_{r,i}) \right]. \tag{2}$$

Node $v_{t,T(i)}$ is the intended transmitter for receiver $v_{r,i}$ and is thus excluded from the inner summation of interference sources. The term $S(v_{t,j}, v_{r,i})$ indicates the relation between the channels used by transmitter $v_{t,j}$ and receiver $v_{r,i}$. Function $S(v_{t,j}, v_{r,i}) = 0$, if $v_{t,j}$ and $v_{r,i}$ are using different (strictly orthogonal) channels. $S(v_{t,j}, v_{r,i}) = 1$, if $v_{t,j}$ and $v_{r,i}$ are using the same channel. In CDMA systems, different channels with non-orthogonal codes may be used, in which case $0 < S(v_{t,j}, v_{r,i}) < 1$.

III. DESIGN PRINCIPLES AND IMPLEMENTATIONS

In this section, we present the three design principles on distributed channel assignment and introduce two example implementations of these design principles.

A. Design Principles

The first and primary principle is the use of a "cross-layer" design approach, where channel assignment is combined with ad hoc routing. This "cross-layer" design principle is motivated by the fact that both the channel assignment algorithm and the ad hoc routing algorithm must be invoked when there is a change in the network topology. In addition, piggybacking channel information in routing control messages can greatly reduce the complexity of channel assignment protocols. For instance, a recently proposed channel assignment algorithm has communication complexity of $O(d^2 \cdot |V|)$, where d is the maximum number of one-hop neighbors that a node can have [4]. This complexity implies that whenever there is a topology change, up to $O(d^2 \cdot |V|)$ messages will be exchanged in the network. Such high complexity prevents the protocol from being implemented in a realistic scenario. However, by utilizing the "cross-layer" design principle, a combined Channel Assignment and AODV (CA-AODV) algorithm has a communication complexity of only $O(1)$, since channel information is carried exclusively through routing control messages.

The second design principle states that channels should be assigned only to active nodes. Before a node has 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. We call a node with a valid route an "active" node. Most existing channel assignment schemes assign channels to *all* nodes in a wireless network, regardless of whether they are active or inactive [2]–[4]. If not all nodes in a network are active at the same time, existing schemes will require more wireless channels than necessary. By assigning channels in an on-demand manner only to active nodes, the resulting protocols can potentially reduce the number of channels required in a wireless network, because the number of required wireless channels is generally proportional to the number of active nodes rather than the number of all nodes.

Finally, distinct channels should be assigned in a way that avoids collisions and interference as much as possible. Many existing channel assignment schemes try to resolve only secondary collisions, whereas primary collisions can also reduce channel utilization. Moreover, interference can corrupt data packets and reduce throughput. Instead of assigning distinct channels to a node and its two-hop neighbors as existing algorithms do, we propose to assign distinct channels to any nodes within a k-hop neighborhood. Neighborhood size k is a user-defined constant that can be set to a value that is appropriate for specific types of networks, modulation, error control coding, and applications. By maintaining distinct channels within a k-hop neighborhood, channel utilization and network capacity can be greatly improved.

B. Implementations

The first and third proposed design principles can be applied to both reactive routing algorithms, such as the AODV routing protocol [6], and proactive routing protocols, such as the Optimized Link State Routing (OLSR) protocol [7]. Because proactive routing protocols are not on-demand, the combined routing and code-assignment schemes are not "on-demand," either. Instead, channel information is maintained for a node's k-hop neighbors at all times. The details of the implementations for different routing protocols may differ, but the essential design principles remain the same. In this paper, we use AODV as an example to illustrate 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.

We previously introduced the combined Channel Assignment and AODV scheme, CA-AODV, in [5]. CA-AODV aims to assign distinct channels to any nodes that are on the same active route and are within k hops of each other. Because the channel assignment algorithm is combined with a routing protocol, the algorithm is not restricted to one-hop neighbors of a node. Instead, a node can obtain channel information for any node k hops away on the same route and then use this information to select a channel for itself. Moreover, CA-AODV does not require any new control messages, just RREQ and Route Reply (RREP) messages.

1) Overview of CA-AODV: CA-AODV, like AODV, has two phases: route discovery and route reply. During the route discovery phase, channel information about a node's k-hop neighbors on 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 $\mathcal{A} \subset \mathcal{C}$, 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 those still available.

CA-AODV avoids channel collisions on a per route basis. Thus, it works well only when there are a small number of active routes that co-exist in a k-hop neighborhood. To improve the performance of CA-AODV, we propose a new Enhanced Channel Assignment and AODV, ECA-AODV.

2) Enhanced Channel Assignment and AODV: ECA-AODV introduces an extra message, the ChannelTaken message, which is broadcast in the route discovery phase. If a node on an established route detects that a new route in the k-hop neighborhood is being set up, it broadcasts a ChannelTaken message that carries its own channel index. The time-to-live (TTL) field of the ChannelTaken message is set to k to make sure that the ChannelTaken message will be broadcast only to the current node's k-hop neighbors. Upon receiving a ChannelTaken message, a node will update its next-hop neighbor table and the available channel set $\mathcal{A}$ by marking the channel indicated in the ChannelTaken message as unavailable. After receiving a RREP message, a node checks to see whether its own channel is in conflict with the channel chosen by an active node. If so, the node will randomly pick another channel from the available channel set $\mathcal{A}$.

Through ChannelTaken messages, channels taken by nodes on established routes can be disseminated to other nodes in the k-hop neighborhood. Therefore, conflicting channels within the k-hop neighborhood can be avoided, provided that the number of available channels is sufficiently large. 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 sending back the RREP message. W_t is related to both k and t_p, where t_p is the per hop propagation time.

The ECA-AODV protocol assigns distinct channels to any active nodes within k hops of the each other, regardless of whether the nodes are on the same route or not. By assigning distinct channels to neighbors within a node's interference range, we not only solve the hidden terminal problem, but also significantly reduce packet errors caused by interference from other transmitters. Therefore, ECA-AODV has better performance than CA-AODV at the cost of higher control overhead, due to the extra control message. The proposed algorithms can work with a multichannel CSMA MAC protocol that separates control channels from data channels. All nodes use the same common channel for control messages, but may use different data channels.

IV. ANALYSIS OF THE ALGORITHMS

In this section, we show that ECA-AODV operates correctly and the number of channels required has an upper bound of $n\cdot(k+1)$, where n is the number of active routes that lie within the k-hop neighborhood of each other. Moreover, we present a numerical example to show that ECA-AODV can give channel assignments similar to the centralized greedy algorithm.

A. Properties of the Algorithms

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

1) Base Case: If there is only one route in the network, it can be easily shown that the number of required distinct channels is $k + 1$.
2) 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 is a new route traversing the k-hop range, the $(n + 1)$-th route can be assigned with $k + 1$ 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 $n \cdot (k + 1)$ distinct channels.

■

Proposition 2: After a new route has been found, each node along the new route is assigned a distinct channel among its k-hop neighbors, provided that the number of available channels is sufficient.

Proof: Under the assumption that the channel assignment procedure is not disrupted by a sudden failure or malfunction of nodes, the channel information carried by control messages will be 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 large enough. Thus, after a new route has been found, each node on the route will have its k-hop neighbors' channel information. Because a node randomly picks its own channel only 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. ■

...	0	1	2	0	1	2	...
...	3	4	5	3	4	5	...
...	6	7	8	6	7	8	...
...	0	1	2	0	1	2	...

Fig. 1. Channel assignment by the greedy algorithm.

...	0	1	2	0	1	2	...
...	5	4	3	5	4	3	...
...	8	7	6	8	7	6	...
...	0	1	2	0	1	2	...

Fig. 2. Channel assignment by ECA-AODV.

V. COMPARISON TO A CENTRALIZED ALGORITHM

Through a numerical example, we show that ECA-AODV can give channel assignments that are similar to the near-optimal Greedy-AODV algorithm. Greedy-AODV assigns channels to nodes on routes that are set up by AODV in a greedy manner, assuming global knowledge of the network topology. In the numerical example, we use a random grid to represent a node distribution, which is a common practice in network simulations. A node is randomly placed on a grid, while within the grid, the node's position follows a random uniform distribution.

Figure 1 specifies the channel assignment made by the centralized greedy algorithm. The first row represents the first established route, the second row represents the second established route, and so on. The number in each grid represents the channel index taken by the node in the grid. Here, k is assumed to be 2, meaning that the greedy algorithm seeks to assign distinct channels to any node within a two-hop range. When the first route is being set up, Greedy-AODV assigns the lowest possible channel indices to the nodes on the first route, subject to the constraint that nodes within the two-hop range must have distinct channels. For easy illustration, in this example, we assume the second route is set up after the first route. Because the lowest three channel indices have already been taken by the first route, which is within two-hop range of the second route, Greedy-AODV assigns the lowest available channel indices to nodes on the second route. Three new channels are assigned to nodes on the third route. Since the fourth route is outside of the two-hop range of the first route, the channels assigned to nodes on the first route can be re-used. Note that this assignment is free of any conflicting channels within any node's two-hop range.

Figure 2 indicates the channel assignment by ECA-AODV. Note that the channels assigned to the nodes on the second and the third routes are in reverse order of the channels assigned by Greedy-AODV. The reason is that the channel assignment is finalized during the route reply phase. Since each node picks the lowest channel index available and the channels are finalized from downstream nodes to upstream nodes, a downstream node will pick a lower channel index than its upstream neighbor. Note that this assignment also has no conflicting channels.

As can be seen from this example, at least in some cases, ECA-AODV can achieve the same performance as the near-optimal greedy algorithm.

VI. SIMULATION STUDY

Using the ns2 simulator, we performed simulations on 25 random scenarios with distinct source-destination pairs. Each random scenario has 64 wireless nodes that are randomly distributed over an 800 m by 800 m area. The two-ray ground path loss model is used and the radio range is assumed to be 180 m. To allow comparison with existing channel assignment schemes, we assume static topologies in our simulations. We measured the average number of conflicting nodes within the interference range as well as the average accumulated interference levels in the network, as defined in Equation (2).

Figure 3 and Figure 4 show the results for the number of conflicting nodes and interference levels, respectively, where the total number of available channels is fixed at $|\mathcal{C}| = 12$ and the neighborhood parameter is set to $k = 2$. Figure 5 shows the effect of neighborhood size k on interference level, where $|\mathcal{C}| = 18$ and $k = 2$ and 3.

Figure 3 reveals that the average number of conflicting channels increases with an increase in the number of active routes. Random-AODV assigns channels randomly to nodes on active routes. Random-AODV has the lowest complexity and can provide a reasonable upper bound on the number of conflicting nodes. CA-AODV has complexity similar to Random-AODV but performs better than Random-AODV. Greedy-AODV is a centralized near-optimal algorithm that seeks to assign distinct channels to any active node within the interfering range. Note that Greedy-AODV can achieve zero channel conflict when the number of active routes is small to medium. The performance of ECA-AODV is always better than that of CA-AODV. Moreover, the performance gap between ECA-AODV and CA-AODV increases when the number of active routes increases. This is because CA-AODV assigns channels to each active route independently through only AODV control messages. Thus, channel conflicts occur more frequently when the number of active routes increases. ECA-AODV can significantly reduce the number of conflicting nodes among different routes through ChannelTaken messages and, thus, the performance of ECA-AODV degrades more gracefully.

Assuming the worst case scenario where all active transmitters transmit simultaneously, we can obtain the accumulated interference level at each receiver, as shown in Figure 4. From Figure 4, we can observe that the accumulated interference levels generated by Random-AODV range from -84 dB to -73 dB, which are higher than the minimum received signal level, i.e. -88 dB, for the 180 m radio range. Again, the performance of CA-AODV degrades sharply when the number of active routes increases, while the performance of ECA-AODV degrades gracefully because ECA-AODV takes all routes within the k-hop range into consideration. If the number of active routes is small in a network, CA-AODV is a good choice because it has minimum control overhead. If the number of active routes is medium to high, ECA-AODV gives better performance at the cost of higher control overhead.

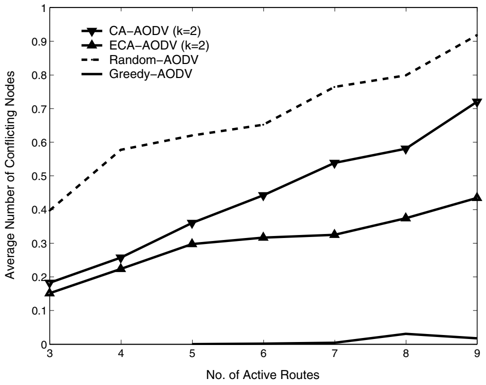

Fig. 3. Average number of conflicting nodes.

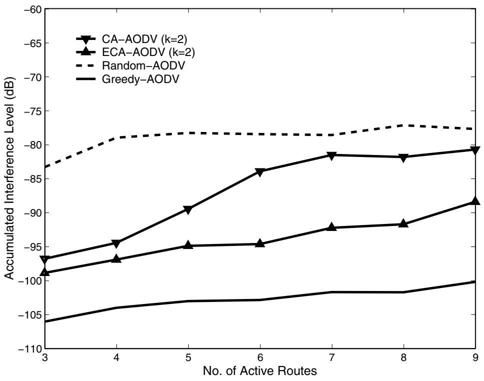

Fig. 4. Accumulated interference levels at each receiver.

Figure 5 shows that when the neighborhood size increases, the performance of ECA-AODV improves. For instance, when $k = 3$, the performance gap between ECA-AODV and Greedy-AODV becomes small. We conjecture that if neighborhood size k and wait time W_t are set to appropriate values, the performance of ECA-AODV can closely approach that of the Greedy-AODV algorithm.

VII. RELATED WORK

Hu's pioneering work [3] examined distributed code assignment for CDMA packet radio networks, including wireless ad hoc networks. Using 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. Four code assignment schemes are defined in [3], namely common code assignment (CCA), receiver-based code assignment (RCA), transmitter-based code assignment (TCA), and pair-wise code assignment (PCA).

Because both decision versions of TCA and PCA are $\mathcal{NP}$-complete, fast heuristic algorithms are studied, including a greedy algorithm, a fan and chain re-coloring algorithm, and

TABLE I
COMPARISON OF CA-AODV AND ECA-AODV TO ALGORITHMS PROPOSED IN [4] AND [3]

Protocols	No. of chann.	Comm. complex.	Comp. complex.	Storage complex.
Centralized greedy algorithm [3]	$d \cdot (d-1)+1$	N/A	$O(d^2 \cdot \|V\|)$	$O(d^2 \cdot \|V\|)$
Distributed channel assignment [4]	$d \cdot (d-1)+2$	$O(d^2 \cdot \|V\|)$	$O(d^2)$	$O(d^2)$
ECA-AODV	$n \cdot (k+1)$	$O(\|V_t\| \cdot k)$	$O(k)$	$O(k)$
CA-AODV	$n \cdot (k+1)$	$O(1)$	$O(k)$	$O(k)$
Random-AODV	N/A	$O(1)$	$O(1)$	$O(1)$

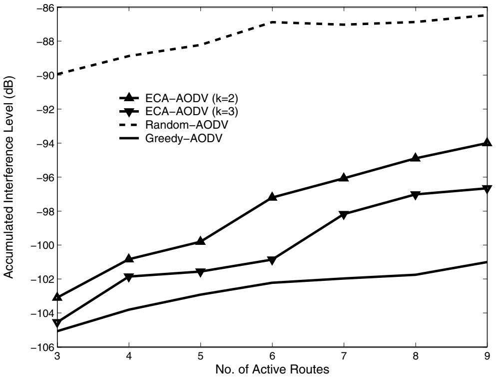

Fig. 5. Impact of neighborhood size on performance.

a sink-tree coloring algorithm. Even though the solutions proposed in [3] are sound, they have high time complexity and/or high communication overhead. Moreover, the schemes do not consider the case when the number of codes is limited and perfect assignment is not possible. Hu's scheme also assumes that each node has knowledge of its neighbors through a routing table. While this is applicable in most cases, assigning channels after the routing protocol converges could incur excessive delay and control overhead.

Garcia-Luna-Aceves and Raju [4] describe a distributed code assignment scheme that assigns distinct channels to a node and its two-hop neighbors. Code Assignment Messages (CAMs) are exchanged along with control messages in the routing protocols. One node's CAM will propagate up to two hops away from the node. If the number of codes available for assignment is at least $d \cdot (d-1)+2$, where d is the maximum degree, i.e., the maximum number of neighbors for any node, there will be no interference after the algorithm converges. However, this algorithm incurs high communication and computation overhead and has high time and storage complexity. Thus, no simulation results are presented in [4].

Most existing heuristic algorithms require more channels than necessary and are too complex to be practical [2]–[4]. In addition, they ignore primary collisions and harmful interference that may corrupt packets and cause retransmissions.

Table I compares several heuristic algorithms in terms of their communication, computation, and storage complexity. For CA-AODV, the control messages are carried solely by routing messages. In addition, channel assignments for each active route are saved along with each routing entry. Therefore, there is no extra communication overhead and little storage overhead. ECA-AODV introduces a ChannelTaken message that is broadcast by nodes on active routes to nodes within a k-hop neighborhood. Thus, the communication complexity of ECA-AODV is $O(|V_t| \cdot k)$. In Table 1, d is the maximum number of neighbors for any node, $|V_t|$ is the total number of active transmitters, n is the total number of active routes within k-hop range of each other, and k is a constant associated with the interference range. Note that in both [4] and [3], only two-hop neighbors are considered.

VIII. CONCLUSIONS

We have presented three design principles for efficient, distributed channel assignment in wireless ad hoc networks. Based on these design principles, we investigated two example protocols, CA-AODV and ECA-AODV. These two protocols exhibit significantly lower communication, computation, and storage complexity than existing channel assignment schemes. Simulation results show that, in some cases, the performance of ECA-AODV can approach that of Greedy-AODV. We also present a numerical example to show that ECA-AODV can give channel assignments that are similar to Greedy-AODV in some cases.

ACKNOWLEDGMENT

We would like to thank Dr. Tao Lin for helpful discussions. Michelle Gong is supported by an IGERT grant from the National Science Foundation (award DGE-9987586).

REFERENCES

[1] S. Xu and T. Saadawi, "Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks?" *IEEE Communications Magazine*, vol. 39, no. 6, June 2001, pp. 130-137.

[2] A. A. Bertossi and M. Bonuccelli, "Code assignment for hidden terminal interference avoidance in multihop packet radio networks," *IEEE/ACM Trans. on Networking*, vol. 3, no. 4, August 1995, pp. 441-449.

[3] L. Hu, "Distributed code assignments for CDMA packet radio networks," *IEEE/ACM Trans. on Networking*, vol. 1, no. 6, 1993, pp. 668-677.

[4] J. J. Garcia-Luna-Aceves and J. Raju, "Distributed assignment of codes for multihop packet-radio networks," in *Proc. MILCOM*, vol. 1, 1997, pp. 450-454.

[5] M. X. Gong and S. F. Midkiff, "Distributed channel assignment protocols: a cross-layer approach," in *Proc. IEEE Wireless Communications and Networking Conference*, 2005, to appear.

[6] C. E. Perkins and E. M. Royer, ed., "Ad hoc on-demand distance vector routing," in *Proc. IEEE Workshop on Mobile Computing Systems and Applications*, 1999, pp. 90-100.

[7] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol," *IETF RFC 3626*, Oct. 2003.