A Coordination Framework for Opportunistic Spectrum Access in Ad Hoc Networks

Haitao Zheng, Jun Zhao
Microsoft Research Asia
Beijing 100080, P.R.China
htzheng, t-juzhao@microsoft.com

Zhensheng Zhang *
San Diego Research Center
San Diego, California, USA
zzhang@ieee.org

Ben Y. Zhao †
Department of Computer Science
University of Santa Barbara, U.S.A
ravenben@cs.ucsb.edu

Microsoft Technical Report MSR-TR-2005-22
Feb. 2005

Abstract

A unique property of open spectrum systems is the heterogeneity in spectrum access. This property provides challenges for MAC designers, including the elimination of common control channels for information exchange. In this paper, we propose a distributed, scalable and efficient coordination framework for Open Spectrum ad hoc networks that takes spectrum heterogeneity into account. The framework is an end-device based approach where devices dynamically select their coordination channel based on local conditions, removing the need for a predefined, commonly available spectrum slice for control traffic. Extensive simulation results show that our distributed channel selection algorithm performs similarly to a centralized near-optimal algorithm, and that under our framework with distributed coordination channels, TCP throughput between devices improves by a factor of 2-4 over a comparable scenario without channel coordination.

Keywords: ad-hoc wireless networks, MAC, coordination, multi-channel, opportunistic spectrum access

1 Introduction

A variety of wireless devices are becoming increasingly ubiquitous, placing additional stress on the fixed radio spectrum available to all access technologies. In order to eliminate interference, current policies allocate fixed spectrum slices to each wireless technology. Its centralized, static nature prevents it from utilizing the dynamic reuse of unused allocated spectrum, resulting in very poor utilization and spectrum holes. Studies have shown that reuse of such "wasted" spectrum can provide an order of magnitude improvement in system capacity [7].

These results provide further motivation for the *Open Spectrum* [2, 6, 9] approach for spectrum access. Enabled by software defined radio (SDR) technology [4, 8], *Open Spectrum* allows users to sense locally available (unallocated or unused by primary users) spectrum ranges and utilize them opportunistically. The unlicensed use of unused spectrum can coexist with legacy spectrum users, increasing capacity and utilization.

A proper MAC protocol (or coordination protocol) is required to exploit the benefits of Open Spectrum. The exchange of coordination and control signals requires a signaling path. Previous works on multi-channel MAC protocols [1, 13, 12] assume homogeneity across devices and the existence of a common control channel. These assumptions do not hold in Open Spectrum systems, where the availability of spectrum ranges fluctuates with location and time, and depends on the device's RF configuration, a property we refer to as *spectrum heterogeneity*. Other approaches [14] statically assign a slice in a licensed band as the dedicated control channel for all devices. Since bandwidth used by control messages increases with device density and spectrum range, this fixed assignment would limit the growth and scalability of these networks. It also mandates a priori assignment (or standardization) of spectrum before deployment, making deployment difficult.

In this paper, we describe the design of a distributed, scalable and efficient coordination framework for Open Spectrum ad hoc networks while considering spectrum heterogeneity. We propose an end-device based approach where each device dynamically selects the coordination channel based on local conditions. This eliminates the requirement of a prede-

*This work was conducted while he was a visiting professor at MSRA.

†This work was conducted while he was a visiting professor at MSRA.

fined, commonly available spectrum slice for control messages, and load-balances control messages across multiple spectrum slices. To the best of our knowledge, this work is the first to provide sufficient coordination across devices to exploit the benefits of Open Spectrum, ensure connectivity and avoid interference without requiring a common channel among devices.

The paper is organized as follows. Section 2 provides background on open spectrum and spectrum heterogeneity, followed by the problem definition and proposed framework in Section 3. Next, we define in detail our approaches to the issues of neighbor discovery and information exchange in Section 4, coordination channel selection in Section 5, and coordination procedures in Section 6. We present simulation and analysis results in Section 7. Finally, we discuss implications of our work and future directions in Section 8 and conclude in Section 9.

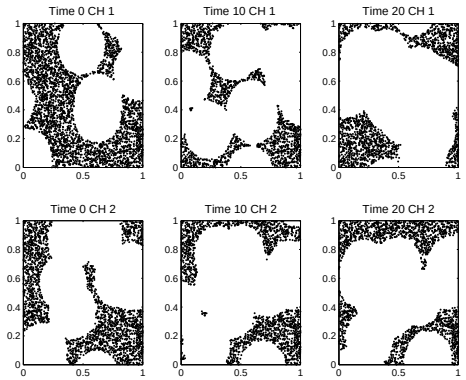


Figure 1: Simulated spectrum availability of secondary users in a 1 km^2 area. We show change in availability at 10 minute intervals, when 30 primary users move randomly with speed between 0-2km/h. Each primary user occupies one channel.

2 Background and Related Works

In this section, we briefly describe the problem of spectrum heterogeneity and related works in the context of Open Spectrum systems.

The *Open Spectrum* approach allows unlicensed users (who we refer to as *secondary users*) to coexist with legacy spectrum users (*primary users*), thereby "creating" new capacity and commercial value from existing spectrum ranges. Secondary users utilize unused licensed bands on a non-interfering or leasing basis based on agreements and constraints imposed by primary users. Secondary users can detect the fixed spectrum signatures or footprints of primaries automatically, through operator-initiated broadcasts, or by accessing a central database. While the goal is to maximize spectrum utilization, an overriding requirement is that secondary users do not interfere with normal operation of primary users. A recent example of this approach is the FCC's recent report on the feasibility of allowing unlicensed devices to operate in TV broadcast spectrum ranges at locations and times when it is underutilized [5]. Devices (secondary users) can detect the presence of a sound carrier in NTSC (analog TV) systems or a pilot tone in ATSC (digital TV) systems, and operate without interfering with TV broadcasts (primary users).

The key to efficient utilization of Open Spectrum is coordination between sender and receiver nodes. Without control, a transceiver needs to scan across channels for incoming messages and select outgoing channels for transmission. Spectrum access based solely on local observations can result in improper use of spectrum opportunities and even harmful interference to neighboring devices. In contrast, coordination allows the transmitter and receiver to quickly agree on the data transmission channel. This not only eliminates the need for a transceiver to constantly scan the spectrum, but also allows peers to negotiate for the most appropriate communication channel(s).

Previous efforts to solve the coordination problem [14] assume that all devices have interference-free access to an *a priori* dedicated control channel, possibly in a licensed band. There are several disadvantages to this approach. First, since the bandwidth requirement of control messages scales with device density and spectrum range, a fixed assignment limits the growth of Open Spectrum systems. Second, it mandates a priori spectrum assignment before deployment, making deployment difficult. In addition, coordination/control channels have been studied in multi-channel MAC protocols [1, 13, 12]. They assume homogenous devices and thus the existence of either a dedicated common channel or a dedicated time segment for control messages to coordinate transmission.

Spectrum Heterogeneity: In Open Spectrum systems, primary users' mobility and traffic variations result in the fact that each device's spectrum availability fluctuates with both location and time, a property we call *spectrum heterogeneity*. For example, Fig. 1 plots an area's spectrum availability to secondary users while occupied by a group of moving primary users. Assuming the spectrum is divided into two channel 1 and 2, and a relatively low mobility rate for primary users, we see that spectrum availability varies significantly over time and across channels. Spectrum heterogeneity is also a conse-

quence of variations in device radio capabilities. For example, a new radio device might have an integrated Ultra Wide Band (UWB) and IEEE 802.11a/b/g interfaces while an older device just has 802.11a.

The lack of available spectrum across a single channel prevents the use of approaches that require a persistent common control channel. In particular, the lack of a common control channel impacts coordination by complicating neighbor discovery and channel coordination. For neighbor discovery, devices need to scan across multiple channels to discover enough neighbors for adequate connectivity, all while minimizing broadcast bandwidth and response time. For channel coordination, sender/receiver pairs need to perform pair-wise negotiations to search for common channels, and to efficiently disseminate such information to other neighbors in the area.

Our work seeks to address these requirements, by providing a framework and algorithms for wireless devices to independently discover neighbors and coordinate communication channels, all in the context of open spectrum systems where persistent common control channels are nonexistent.

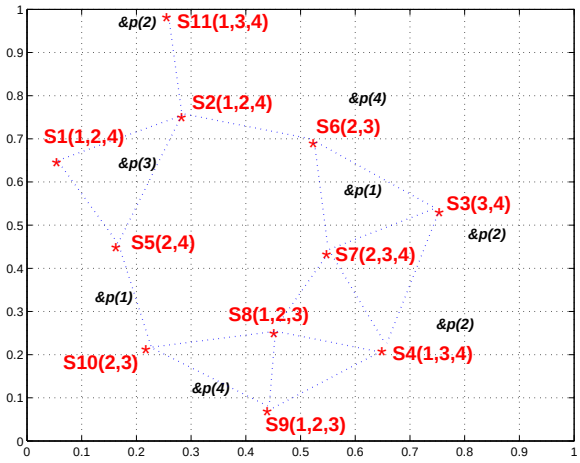


Figure 2: A sample topology of Secondary Users (SE: S1-S11) and Primary Users (PR: &p(d)), where $Si(a, b, c)$ represents secondary users i, with a, b, c as available channels), $p(d)$ represents a primary user occupying channel d. A line connects two secondary users if they are within transmission range of each other.

3 Problem statement and Proposed Framework

In this section, we define the context and assumptions of our work. We consider a network consisting of both primary users (license holders) and secondary users (unlicensed users). Spectrum licensed to primary users are accessible to secondary users under the constraint that secondary users do not interfere with transmissions of primary users. The collection of opened spectrum forms a spectrum pool, and is divided into non-overlapping orthogonal channels, each associated with a channel access technology, *e.g.* CSMA, TDMA, etc. Assuming that they can detect the presence of primary users [1], each secondary user keeps a list of channels unoccupied by primaries.

We assume that transceivers on secondary user devices are frequency agile (*i.e.* able to transmit or receive in different spectrum slices if their frequencies are known prior to transmissions), but not necessarily wideband tunable (*i.e.* can scan through a wide spectrum range and detect frequencies of active transmissions in real time). Therefore, senders and receivers need to negotiate channels for communication. We assume that at any given time, each transceiver can only communicate (send/listen) through one channel with one neighbor, (*i.e.*, transceivers are half-duplex). Each device selects a channel and transmit power before each communication. Although transmit power control can be employed to balance interference range and connectivity, we assume that each device transmits at a fixed power to provide a short-range transmission.

Fig. 2 shows a sample network topology that consists of 8 primary devices and 11 secondary devices. The spectrum is divided into 4 channels and each primary device randomly selects a channel to use. Available channels at secondary devices are derived according to the channel occupation of primary users.

Spectrum heterogeneity, as illustrated in Fig. 2, prevents the use of a common control channel. Without a common control channel, secondary users cannot communicate and utilize the available spectrum efficiently. Efficient use of underutilized spectrum requires a distributed protocol to find coordination channels for each secondary device. The number of coordination channels used by each device should be minimized to reduce signaling overhead and frequent hopping between channels.

Given the assumptions above, we *propose a scalable, distributed and efficient coordination protocol for Open Spectrum enabled ad hoc networks with spectrum heterogeneity.* The protocol enable automatic neighbor discovery, distributed coordination channels selection and efficient spectrum utilization. As topology changes or nodes join/leave the network, devices should self-organize and dynamically adjust coordi-

[1] Secondary users can perform spectrum analysis or listen to a spectrum server's broadcast of primary users and their channel usage.

nation channels. This protocol eliminates the need of a fixed, widely accessible coordination channel, and reduces congestion by distributing load across coordination channels.

To address these issues, we propose a MAC layer framework consisting of the following modules:

- *Neighbor Discovery and Information Exchange*: Devices announce their existence via beacons to their neighbors. Using broadcast beacons, they also exchange information with neighbors on channel availability, coordination channels and associated information.
- *Distributed Coordination Channel Selection*: Devices dynamically select coordination channels(s). This is the core component of the framework.
- *Coordination Protocol*: Once coordination channels are selected, users utilize them according to the specified protocol.
- *Spectrum Coordination*: Devices coordinate transmissions on data channels through local coordination channels.

In the following sections, we will describe each component in more detail.

4 Neighbor Discovery and Information Exchange

Neighbor discovery is the process where devices advertise their existence to their neighbors and listen to advertisements to learn about them. An active device periodically broadcasts an advertisement to the neighborhood, announcing its availability and some associated information, *e.g.* the existence of primary users. When a device is powered on, it first performs a scan to discover neighbor advertisements, and sends out a neighbor solicitation to ask for immediate advertisements [2]. In general, beacons are transmitted during Beacon Transmission Time (BTT). The BTT is a regularly scheduled short time slice dedicated for beacon transmissions, and known to every device. The beacon transmission interval (T_{beacon}) refers to the time interval between two consecutive BTTs. We assume that devices are time synchronized using techniques from literature [3, 11, 12].

In this work, we assume a true ad hoc network without central access points. Each device broadcasts beacons to advertise its existence. In previous systems, devices sent all beacons across a common control channel accessible to all. Because of spectrum heterogeneity in Open Spectrum systems, however, each device needs to broadcast and listen to beacons on multiple channels to ensure neighbor discovery.

To minimize the resulting bandwidth overhead, devices can increase the beacon transmission interval T_{beacon} by a constant factor C, where C is the number of channels in the spectrum pool. A device then staggers or rotates its beacons across different channels separated by intervals of T_{beacon}, such that the resulting beacon rate across all channels remains $\frac{1 beacon}{T_{beacon}}$. For example, let's say a device D beacons once every second on a common control channel. In an open spectrum system with 5 channels, it would beacon on each channel at 5 second intervals, starting with channel 1 at time 1 second, channel 2 one second later, channel 3 one second later, and so on.

Beacons contain information on channel availability, subscribed coordination channel(s) and information (traffic load, etc.) about each coordination channel. Since each device can only sense properties of its immediate area, decisions made solely on local information can be inaccurate. Beacon broadcasts allow devices to share common information such as channel conditions with neighbors, providing each device with more accurate information about a wider area, resulting in more accurate identification of spectrum opportunities and an improved ability to utilize them.

When a device joins the system, it first scans and listens to multiple channels for beacons. It gathers local information about nearby primary users, a list of neighbor devices, their available channels and associated coordination channels. It also notes the scheduled time slice during which neighbor devices will listen to each particular channel. The device uses this information to chooses its coordination channel(s) (see Section 6). If no neighbors are detected within a timeout period, the device assumes no neighbors are in transmission range, and either moves to other location or waits for other devices to join.

5 Coordination Channel Selection

Now we are aware of wireless neighbors within our transmission range, the next step is to communicate amongst secondary devices to choose a set of commonly available coordination channels. First, we propose a theoretical model for solving the problem in a centralized manner over a fixed topology. Then, we describe a distributed coordination channel selection algorithm and illustrate its usage through examples.

5.1 Theoretical Model for a Fixed Topology

For a fixed network topology, we can use a centralized algorithm to determine the optimal coordination

[2] IEEE 802.11 uses a similar beacon and beacon request scheme to exchange information between an access point and devices in its range.

channel assignment, where the number of coordination channels is minimal. We now reduce the centralized problem to a variant of the graph coloring problem by mapping channels into colors.

We define a bidirectional graph $G = (U, E, C, L)$ where U is a set of vertices denoting the secondary devices. For each vertex $v \in U$, a set of local colors represent available channels at the device. M denotes the size of U. C represents the collection of colors (channels), and L the color availability vector (the set of colors that each node can be assigned), a matrix of $|U|$ by $|C|$:

$$L = \{l_{u,i} | l_{u,i} \in \{0,1\}, u \in U, i \in C\} \quad (1)$$

E is the set of undirected edges between vertices representing connectivity between two vertices. That is, if two vertices are within transmission range and they share at least one common color/channel in L, then they are connected by an edge and must share one common color in their color assignment:

$$E = \{e_{u,v} | e_{u,v} \in \{0,1\}, \forall u, v \in U\} \quad (2)$$

Note that $e_{u,v} = e_{v,u}$, and if $e_{u,v} = 1$, then $\sum_{i \in C} l_{u,i} l_{v,i} \geq 1$. In the following, we assume that U is a connected graph [3]. We define a coordination channel assignment at vertex u as:

$$A_u = \{a_{u,i} | a_{u,i} \in \{0,1\}, a_{u,i} \leq l_{u,i}\} \quad (3)$$

Given graph G, we color each vertex using a number of colors from its color list (in L), such that if an edge exists between two vertices, then they share at least one color in their assigned colors. That is, each vertex u needs to connect to all of its neighbors:

$$R(A_u) = \prod_{v \in U, e(u,v) > 0} e(u,v) \sum_{i \in C} a_{u,i} \cdot a_{v,i} > 0 \quad (4)$$

We refer to this as the connectivity constraint.

The goal of the graph coloring problem is to minimize the average number of colors used by each vertex subject to connectivity constraints. It is equivalent to minimizing the number of coordination channels used by each device in order to reduce signaling overhead and frequent hopping among channels. For example, if all vertices share a common channel/color, then the optimal solution is to color all vertices with that color. With spectrum heterogeneity from open spectrum systems, this problem appears to be NP-hard.

[3] If the graph can be reduced into a set of disconnected sub-graphs, then the optimization can be applied independently at each sub-graph.

In the following, we present a centralized greedy algorithm that performs well for common topologies. The algorithm consists of the following three sub-modules.

Subgraph Generation: We find and assign "must-have" colors. For each edge, if two associated vertices only share one color on their color list, assign that color to both vertices. We then segment graph G into sub-graphs associated with each color, *i.e.* a color c sub-graph includes only vertices in G that have the color c on its available list, and the associated edges.

Greedy Color Assignment: We incrementally "process" subgraphs starting with the largest. We start with the subgraph S_G containing the most number of edges and assign all vertices S_G's color. We then examine the remaining subgraphs and remove from them any common edges shared with S_G. This process is repeated on the next largest subgraph, until all subgraphs have been processed, or contain no edges.

- Based on G, define a connectivity graph $T = (U, E)$.
- Step 1: For each color c, we define a colored sub-graph $T_s(c) = (U_c, E_c)$ where

$$\begin{aligned} U_c &= \{u \mid l_{u,c} = 1, u \in U\} \\ E_c &= \{e_{u,v},\ u, v \in U_c\} \end{aligned} \quad (5)$$

- Step 2: Compute $N(c) = \sum_{u,v \in U_c, u<v} e_{u,v}, c \in C$ as the number of edges in sub-graph c. Find the colored sub-graph with the largest number of edges, and assign the associated color to each vertex in the sub-graph.

$$\begin{aligned} c^* &= arg \max_{c \in C} N(c) \\ a_{u,c^*} &= 1,\ \forall u \in U_c^* \end{aligned} \quad (6)$$

  If two sub-graphs have the same number of edges, select the sub-graph with the minimum number of vertices, as it has less number of isolated groups.

- Step 3: Reduce the connectivity graph and sub-graphs to exclude the edges that have been covered by sub-graph c^* *i.e.*

$$\begin{aligned} T &= T \setminus H(c^*) \\ H(c) &= H(c) \setminus H(c^*), \forall c \in C, N(c) > 0 \end{aligned} \quad (7)$$

  if $T = \phi$, then stop, otherwise go to step 2.

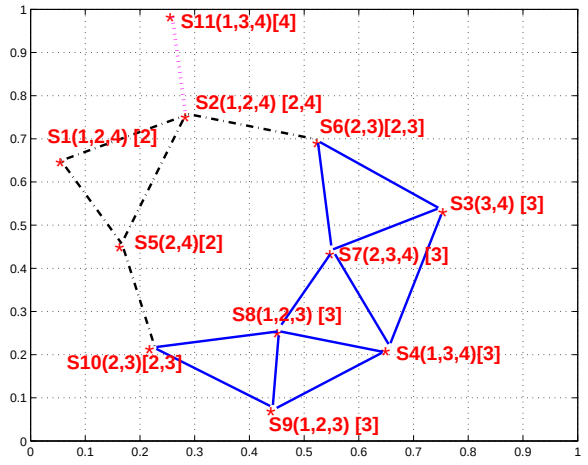


Figure 3: An example of Coordination Channel Selection where $S_i(a,b,c)[a,c]$ represents secondary user i, with a,b,c as available channels and a,c as coordination channels.

We note that constraints can be incorporated into the assignment algorithm. For example, if there is a upper bound on the load of each channel, *e.g.* one vertex can not use the same color to connect to more than χ_n vertices, we need to prune the color subgraphs so that each vertex has no more than χ_n edges.

Color Prune: We prune unnecessary colors from node assignments. For each color assigned to vertex v, v generates a "color list" of all vertices connected to v by an edge and who also share this color assignment. If the "color list" for color c_1 is a subset of c_2's color list, then we remove the color c_1 from all nodes in its color list. This is performed iteratively for each vertex in the network.

Fig. 3 illustrates the color (coordination channel) assignment for the example in Fig 2, using the above centralized greedy algorithm. After generating subgraphs, the algorithm first selects channel 3 as the subgraph covering the most number of edges (11), channel 2 as the subgraph with largest number of left over edges (5) and finally channel 1 to cover the final edge between S2 and S11. From a network perspective, devices in transmission range (neighbors) want to share a common coordination channel, and self-organize into a virtual group for local coordination. Virtual groups likely overlap, since devices at the group boundaries will to subscribe to multiple coordination channels to ensure connectivity to their neighbors. These devices behaves like bridges between groups and ensure global connectivity.

5.2 Distributed Coordination Channel Selection

Our centralized approach to channel selection requires a central controller with knowledge of the entire network topology, and is thus unsuitable for decentralized dynamic networks (*e.g.* ad-hoc networks). We now propose a distributed algorithm where each device selects its coordination channel(s) dynamically. By eavesdropping on beacon broadcasts, devices can hear their neighbors' channel availability and chosen coordination channel(s). To ensure adequate connectivity, each device wants to choose the channel that is used by the largest number of neighbor devices. Devices need to balance the tradeoff between maximizing connectivity and reducing the overhead and complexity of managing multiple coordination channels. The selection criteria are as follows:

- Connectivity: choose channel used by the largest number of active neighbors.
- Transmission range: for equally popular channels, choose the one with higher transmission range [4], thus maximizing the chance that the signal can reach any new devices.
- Load-balancing: Each device announces the load of its coordination channel in beacon signals, allowing other devices to avoid highly loaded coordination channels.

At network initialization, each devices performs discovery to get a list of its neighbors and their channel availability. Each device then iterates through the following steps until a coordination channel is established with each neighbor. We begin by putting all neighbors on our "todo list."

1. Connectivity broadcast: We define a channel's connectivity level as the number of neighbors that can be connected through this channel. Using knowledge of available channels at neighbors on our todo list, each device finds the local channel with the highest "expected" connectivity, and broadcasts to those neighbors the channel ID, its connectivity level and a random number between 0 and 1.
2. Channel selection: After recording beacons from its neighbors on its todo list, each device selects the channel specified by the largest number of those neighbors. To break ties, it chooses the

[4]The channels can differ in transmission ranges due to frequency and power constraints of the assigned spectrum.

channel that has the beacon with the largest random number inside. Each device then broadcasts its selection. Once it has learned of its neighbors who have joined this coordination channel, the device removes those neighbors from its todo list.

3. "Must Have" channel selection: If there is only one channel common to two neighboring devices, then these two devices will choose this channel as coordination channel.

4. Repeat if needed: repeat steps 1-3 until reasonable connectivity is established.

In each iteration of the above algorithm, a device chooses a common coordination channel with a subset of its neighbors. It removes them from its list, and repeats the process until a coordination channel has been established with enough neighbors to ensure connectivity. We can use the results of previous work [15] to determine when reasonable connectivity has been established. Fig. 4 illustrates the procedures of distributed coordination channel assignment for our example in Fig 2.

Devices should adjust coordination channels to adapt to changes in the network, including changes in topology or join/leaves of primary and secondary users. For system stability, these adaptive steps should occur at large time intervals regardless of the frequency of network change events.

When a device joins the network, it performs neighbor discovery to get the channel availability and coordination channels of its neighbors. From the list of coordination channels used by neighbors, it chooses the uncongested channel used by the largest number of common neighbors. If no reasonable choices exist, it chooses one of its available channels and negotiates with its neighbors via beacon broadcasts. When a device adds a channel to its coordination channel list, it performs the color prune procedure defined in Section 5.1 to remove redundant channels.

When a primary user appears and wants to use one channel, all the impacted devices must release the channel within a predefined time frame. Devices using this channel for coordination can use this time to negotiate a new coordination channel. One solution is to choose one of the coordination channels currently used by neighboring devices. This is equivalent to joining a neighboring virtual group. If no feasible virtual groups exist, they can use the channel connectivity broadcast to select new coordination channels and form new virtual groups.

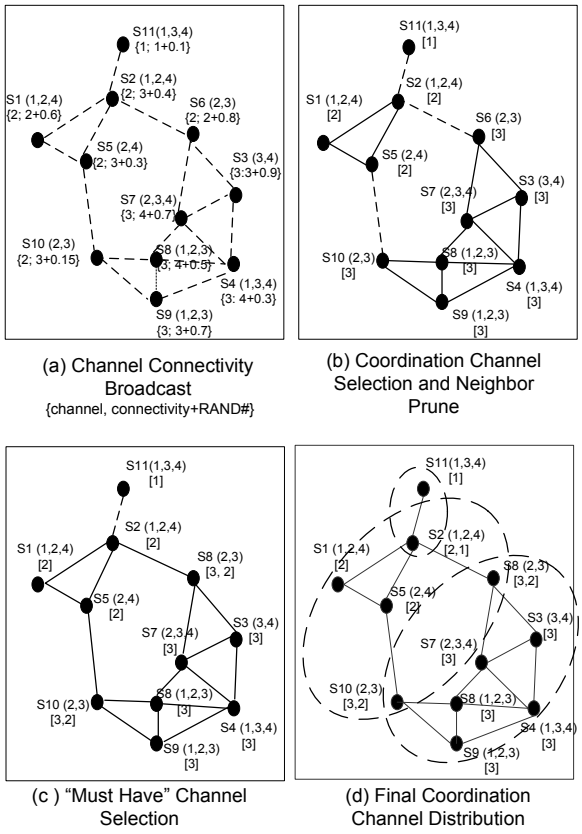


Figure 4: An example of distributed coordination channel selection. In (a), n,m+x refers to n: intended coordination channel, m: the connectivity level of the channel, x: a random number between 0 and 1.

6 Coordination Protocol

Once network devices have selected their coordination channels, they need to determine when and how to utilize the coordination channel to control data transmission. In this section, we examine this problem in two phases: how nodes align their time slots to send and listen on the same coordination channel, and how they negotiate to schedule data transmissions.

6.1 Coordination Channel Alignment

In a realistic scenario, some devices in the system must actively utilize multiple coordination channels to maintain network connectivity. To coordinate control information exchange across multiple channels, these devices must schedule their transceivers to listen on each coordination channel at the same time slice as other devices on that coordination channel. In this section, we present a simple approach to this complex scheduling problem. We are develop-

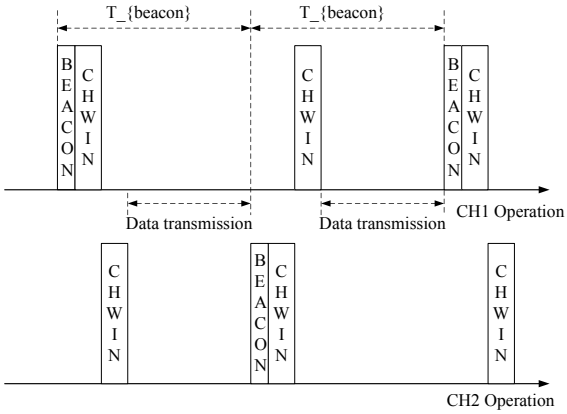


Figure 5: An example of CHWins for a system using 2 coordination channels.

ing more scalable algorithms for coordination channel alignment as part of ongoing work.

A device with k ($k \geq 2$) coordination channels belongs to k different virtual groups (Section 5.2). While we can dedicate the channel to only control traffic [13], this can severely limit the number of channels available for data transmission when the number of channels is small. This is particularly of concern to devices with multiple coordination channels.

Instead, we propose to use a channel negotiation window approach similar to [12]. Each coordination channel K has a time slice, its *channel negotiation window (CHWin)*, during which devices in its virtual group use K to coordinate and schedule data transmissions. Each device uses collision resolution protocols such as CSMA/CA to resolve any collision within group K as in [12].

A device that utilize multiple coordination channels (and thus belong to multiple virtual groups) must communicate on CHWins for each of its coordination channels. Since it can only listen to one channel at a given time, CHWins for its coordination channels must not overlap in time. We propose a simple approach to guarantee this non-overlap property for all devices: the scheduled order of CHWins is predefined and known to all devices. One simple order is to schedule all CHWins in sorted order by channel name or number. For example, in a system with three channels A, B and C, the CHWin order is A, B, C. During the first time slice after every beacon, devices in A's virtual group send/receive coordination messages on channel A. During the next time slice, all devices in B's virtual group coordinate on channel B, followed in the next slice by C's virtual group on channel C. This ensures that no combination of coordination channels overlap in their CHWin time slices.

The basic scheme is shown in the first T_{beacon} time period of Fig. 5, where the CHWins of 2 channels is followed by a long time slice designated for data transmission. One result of this design, however, is that in each beacon period T_{beacon}, the channel with the lowest sort valued name (*e.g.* A) will always have its CHWin scheduled first after the beacon. This means that for a device m utilizing two or more coordination channels (A and B), other devices on channel A can always make a reservation with m for the data transmission segment, making m unable to listen to another channel during the data transmission segment. Over time, scheduling of data will unfairly favor devices using coordination channel A over devices using coordination channel B.

To impose fairness, we modify the design so that at every beacon interval, the first CHWin is assigned to the channel that the beacon is sent out on. Recall from Section 4 that beacons are sent out on a rotating order of channels. By scheduling the first CHWin on the beaconing channel, we ensure each channel gets a fair share in a rotating list of channels, where the rest of the CHWins remain in sorted order. For example, where the basic scheme would schedule CHWins in order (A, B, C) in every beacon period, the modified fair scheme would rotate: (A, B, C), followed by (B, C, A), followed by (C, A, B). Fig. 5 shows this fair scheduling scheme for a system with 2 channels. Note that each beacon period changes the ordering of CHWins.

In a system with a total of N channels, for each coordination channel there are $N - 1$ unused time slices. To improve efficiency, those time slices can be used for control information exchange within the group or for discovering new nodes who may want to join the group. Devices within the group can listen for channel activities to get information on other neighbor's transmissions and detect the activity of primary users in the neighborhood. Detailed discussion of these activities are beyond the scope of this paper.

We recognize this scheme is not optimal in its time-slice allocation, and bandwidth overhead scales linearly with the number of channels. We discuss this limitation and more efficient algorithms in Section 8.

6.2 Spectrum Coordination

Once the coordination channels are selected and the associated CHWins are planed, devices can negotiate data transmissions. Devices who intend to communicate can negotiate the choice of data channel on

the coordination channel, while others monitor the coordination channel to collect information of channel status in different data channels. This is helpful to characterize spectrum opportunities and regulate channel access to avoid harmful interference. In this work, we use a procedure similiar to that of [12], where each pair of transmitter/receiver negotiates the data channel to use and broadcast the information through coordination channel. The detailed procedure is as follows:

Based on the CHWin design, each transmitter sends out a CHI-REQ packet with a preferable channel list, that indicates quality/priority of the data channel that it would like to use. After receiving a CHI-REQ packet, the intended receiver chooses one data channel from the list by integrating the priority of transmitter and receiver. It then informs the transmitter about its decision through an CHI-ACK packet in the same coordination channel. The transmitter then confirms with an CHI-CFM to the receiver with the decision. Both transmitter and receiver set their preferable channel list to include only the selected channel. Neighbors who hear CHI-CFM packet can update their record on channel conditions and modify their preferable channel list to facilitate future negotiations. In particular, each device can count the number of CHI-CFM that it has received in the current CHWin to determine the load on each data channel and update its quality/priority measure on each channel. This can be integrated with other information obtained through coordination information exchange to provide a proper acquisition of channel status.

After selecting the data channel, each transmission pair can invoke the MAC protocol specified by the chosen channel to access channel and transmit packets. For example, IEEE 802.11 involves random backoff, RTS/CTS handshaking to reduce collision among multiple transmissions. It should be noted that those MAC overhead are distributed into each data channel, rather on coordination channel. This approach reduces the coordination overhead and provides better system scalability.

7 Experimental Results

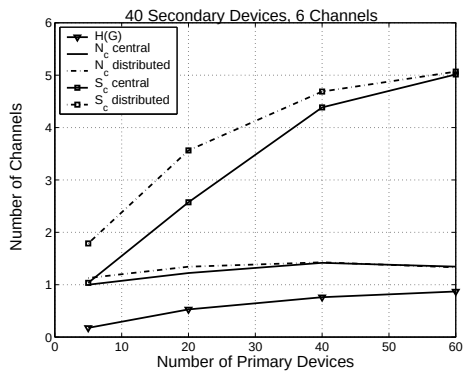
7.1 Coordination channel selection

We investigate the statistical performance of coordination channel selection, based on an immobile system consisting multiple primary and secondary devices in a unit one by one area. The spectrum is divided into 6 channels and each primary device randomly selects a channel to use. Here for simplicity, we do not assume any impact of interference between primary devices. We further assume that each primary device has an interference sensitive range of R_p, that is, if a secondary device is within R_p distance from a primary device, then the secondary device can not use the channel that the primary device occupies. We also assume that if two secondary devices are within R_s distance from each other, they can interconnect.

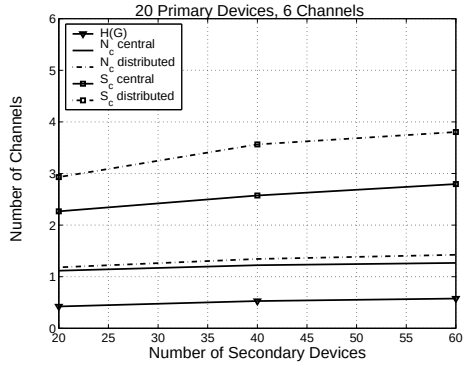
We randomly place primary and secondary devices, and derive the available channel at each secondary device based on primary users' channel occupation. We evaluate the performance of coordination channel assignment using centralized and distributed algorithms described in Section 5. The results are averaged over 2000 device placements. The statistics are the average number of coordination channels assigned to each secondary user (N_c) and the average total number of coordination channels required to coordinate the system (S_c). S_c reflects the average number of virtual groups in the system. We also define an indicator of heterogeneity $H(G)$ as the probability of a channel in each vertex's available list that is not shared by all of its neighbors. Apparently, if the channel availability at each secondary user is the same, $H(G) = 0$.

Fig. 6 illustrates the statistical results for the simulated system with different number of primary and secondary devices. Increasing the number of primary devices or the number of secondary devices increases the level of spectrum heterogeneity, and consequently S_c and N_c. It is interesting that N_c is relatively small (1-1.5). This is mostly due to that channel availability is location dependent. Unless the primary sensitivity range R_p is significantly smaller than secondary device's transmission range R_s, neighboring devices are likely to have one or two channels in common. Once the value of R_p and R_s are defined, S_c is relatively insensitive to the number of primary devices and the number of secondary devices. On the other hand, the value of S_c strongly depends on the global topology, and hence quite sensitive to the variation of the number of primary and secondary devices.

We observe that distributed algorithm produces similar N_s value as that of centralized algorithm. However, there is still visible difference in terms of S_c. This is again because S_c depends strongly on the global topology while N_c only depends on local conditions. Hence, centralized algorithm which operates based on knowledge of global topology and device condition produces superior performance in S_c, but less visible improvement in N_c.



(a)



(b)

Figure 6: Coordination channel statistics assuming $R_p = 0.2$, $R_s = 0.3$ and 6 channels.

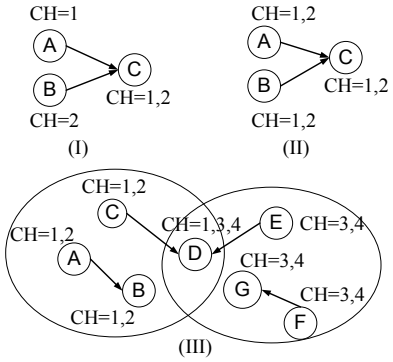


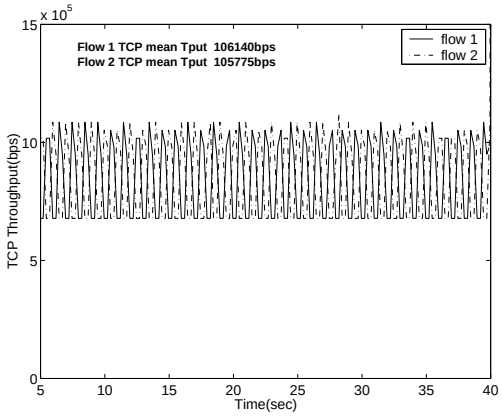
Figure 7: Simulation Topologies

7.2 Spectrum coordination using CSMA protocol

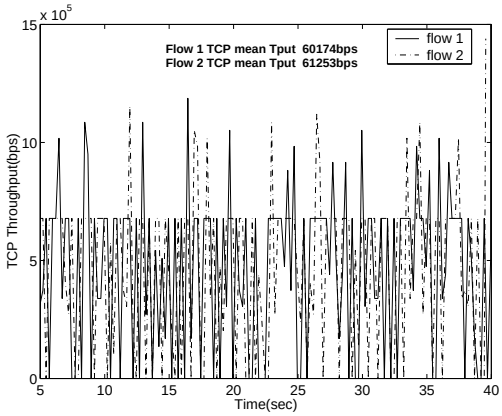
We compare the performance of our proposed coordinated system that uses coordination channel to negotiate spectrum access, to a uncoordinated system where each secondary device randomly hops among available channels. After primary devices have selected the channel to use, the channel availability at each secondary device are derived, and can be characterized by a topology. We modify standard NS code of IEEE 802.11 single channel system to include multiple channels and coordination CHWins. Data transmission at each channel is at 11Mbps data rate. The CHWin interval for coordinated system is set to 100ms with CHWin length of 5ms, and the hopping interval for uncoordinated system is set to 100ms for a fair comparison. We conduct extensive studies on various topologies. Due to space limit, we focus on three fixed topologies specified in Fig. 7. For topologies I and II, we assume all the traffic flows are large file transfer running on TCP, and use TCP throughput as performance metric. For topology III, all the flows are streaming video with constant bit rate running on UDP, and packet delay is used for performance evaluation.

Topology I consists of three nodes without any common channel, so that device C has to use channels 1 and 2 to coordinate with A and B separately. Hence, flows from A to C, and B to C are multiplexed in time. This can be observed in Fig. 8(a). On the other hand, in a uncoordinated system, transmissions are constantly interrupted as C hops between channel 1 and 2, which produces throughput and delay variations. When data channels of transmitting and receiving devices mismatch, all the packets sent to the receiver are lost. However, as the transmitter can not differentiate the loss due to channel mismatching, transmission errors, or confliction, it simply retransmits the lost packets. This increases the delay variance of TCP packets. Overall, we see that coordination produces 175% of throughput improvement.

In topology II, devices A, B and C select channel 1 as coordination channel. In coordinated system, during each CHWin, both flows are scheduled in the same data channel. For example, A schedules with C to use channel 1 as data channel, and C update its preferable channel list to only include channel 1(as it will use channel 1 for data channel in the following data transmission interval). When B negotiate with C, C chooses channel 1. Hence, both flows share channel 1 and resolve collision through CSMA protocol. This can be observed from TCP throughput in Fig. 9(a)). On the other hand, uncoordinated system suffers from many transmit/receive channel mismatch as A, B and C randomly hop among channels 1 and 2. Large delay jitter leads to frequent TCP timeout, *e.g.* flow 1's TCP timeout at 12 sec and resumes at 26 sec (Fig. 9(b)). In this case, coordination produces as much as 4 times of throughput improvement. Notice that the coordinated system for this topology is similar to that in [12]. We want

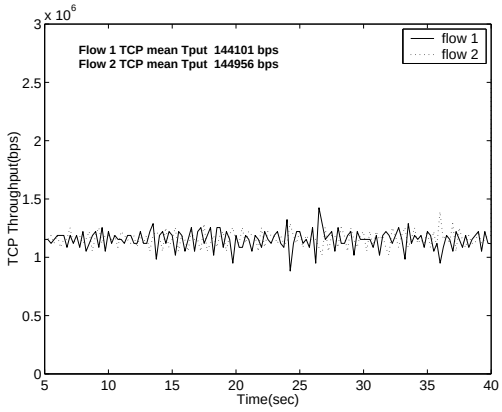


(a) Coordinated system

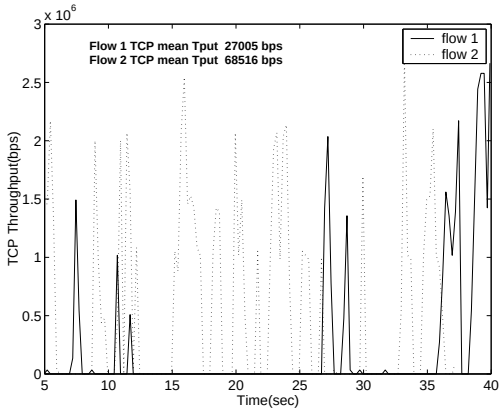


(b) Uncoordinated system

Figure 8: TCP throughput of topology I's traffic flows



(a) Coordinated system



(b) Uncoordinated system

Figure 9: TCP Throughput of topology II's traffic flows

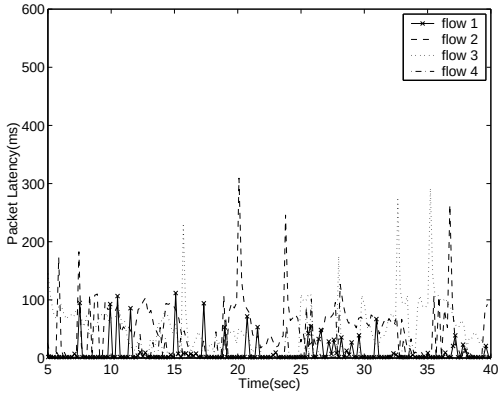
to emphasize the gain of scheduling two flows with the same destination on the same data channel, and its impact on TCP performance.

In topology III, we examine a fully connected system consisting of 2 virtual groups with one overlapping device. We assume UDP flows with arrival rate of 100 packets per second, each packet being 512bytes long. All the flows are conflicted and hence can occur concurrently if using different channels. Within each group, coordination allows conflicting transmissions to take place on different channels, *e.g.* flow A to B uses channel 2 as flow C to D uses channel 1. Inter-group coordination, performed by D, resolves conflict between flow C to D and E to D. Results in Fig 10 show that coordination reduces packet delay significantly.

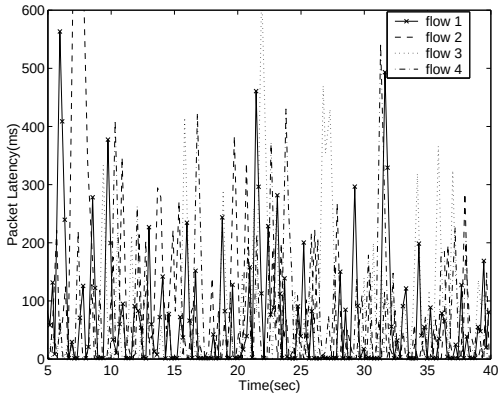
8 Discussion and Future Work

Scalability: In this paper, we propose a simple alignment scheme to coordinate CHWins. In its current format, the coordination overhead scales linearly with the number of channels. This design limits the scalability of the system. However, as we observed, the number of coordination channels used in an area (S_c) is usually less than the total number of channels, it is feasible to develop more efficient algorithm to reduce the number of CHWin time slices that each device should tune to. For example, the CHWin time of a coordination group (virtual group) can be selected on-demand. Before setting up a new virtual group, devices should use knowledge about neighboring virtual groups to select the position of beacon and CHWin to avoid overlapping/interference with their neighboring virtual groups. The positions of CHWin of different channels should also rotate to provide fairness in channel negotiation. We are currently investigating this problem.

Extension to coexistence systems: In this paper, we assume that each channel is associated with a predefined channel access scheme, *i.e.* CSMA, TDMA, etc. Our scheme can also be extended to the co-



(a) Coordinated system



(b) Uncoordinated system

Figure 10: Packet delay of of topology III's traffic flows

existence problem where channels are shared by devices that are associated with different and mutually interfering access technologies. An example will be coexistence of 802.11 and 802.16 systems in the unlicensed band. In the current setting, two devices can only communicate if they have the same technology. Therefore, two pairs of transmissions with different technologies will interfere with each other. There have been proposals on using out-of-band channel(s) as coordination channel [10]. However, it requires each transceiver to access multiple channels concurrently. Our proposed scheme can be extended to provide coordination among different technologies without the use of any out-of-band channels. We propose to implement a common transmission/access technology during beacon and CHWin periods on each device. Coordination aims to schedule transmissions involving different technologies onto different channels to avoid collision. We plan to investigate this problem in a future paper.

9 Conclusion

In this paper, we present a distributed, scalable and efficient coordination framework to explore/utilize the unused bandwidth in Open Spectrum ad hoc networks while considering spectrum heterogeneity. Under the framework, each device dynamically selects the coordination channel based on local conditions, eliminating the need of a common channel for control messages. We present both a centralized formulation for coordination channel selection as well a distributed algorithm. Simulations show that our distributed algorithm performs extremely well compared to the centralized algorithm. Extensive network simulations also show that system throughput with control channel coordination increased by a factor of 2–4 compared to no coordination. We also discuss the limitations and extension of the proposal and are currently developing new schemes to address these issues.

Acknowledgement

We would like to thank G. Yang for his help in NS simulations.

References

[1] Adya, A., Bahl, P., Padhye, J., Wolman, A., and Zhou, L. A multi-radio unification protocol for IEEE 802.11 wireless networks. Tech. rep., Microsoft Research, 2003.

[2] Berger, R. J. Open spectrum: a path to ubiquitous connectivity. *ACM Queue 1*, 3 (May 2003).

[3] Elson, J., Girod, L., and Estrin, D. Fine-grained network time synchronization using reference broadcasts. In *Proc. of OSDI* (Dec. 2002).

[4] FCC. Facilitating opportunities for flexible, efficient and reliable spectrum use employing coginitive radio technologies. FCSS 03-322.

[5] FCC. Notice of inquiry: Additional spectrum for unlicensed devices below 900 mhz and in the 3 ghz band, Dec. 2002. ET Docket No. 02-380.

[6] FCC spectrum policy task force. `http://www.fcc.gov/sptf`.

[7] McHenry, M. Spectrum white space measurements. *New America Foundation Broadband Forum* (June 2003).

[8] Mitola III, J. Wireless architectures for the 21st century. `http://ourworld.compuserve.com/homepages/jmitola`.

[9] Powell, M. K. Broadband migration iii: New directions in wireless policy. Remarks at the Silicon Flatirons Telecommunications Program, Oct. 2002.

[10] Raychaudhuri, D. Adaptive wireless networks using cognitive radios as a building block. MobiCom2004 Keynote Speech, September 2004. Philadephia.

[11] Römer, K. Time synchronization in ad hoc networks. In *Proc. of MobiHoc* (Oct. 2001), ACM, pp. 173–182.

[12] So, J., and Vaidya, N. Multi-channel MAC for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver. In *Proc. of MobiHoc* (May 2004), ACM, pp. 222–233.

[13] Wu, S. L., Lin, C. Y., Tseng, Y. C., and Sheu, J. P. A new multi-channel (mac) protocol with on-demand channel assignment for multi-hop mobile ad hoc networks. In *Proc. of I-SPAN* (2000), pp. 232–237.

[14] XG working group RFC, the XG vision and the XG architecture. `http://www.darpa.mil/ato/programs/XG`.

[15] Xue, F., and Kumar, P. The number of neighbors needed for connectivity of wireless networks. *Wireless Network 10*, 2 (March 2004), 169–181.