# Distributed Coordination in Dynamic Spectrum Allocation Networks

Jun Zhao
Microsoft Research Asia, Beijing, P.R.China
junzhao@microsoft.com

Haitao Zheng
Univ. of California, Santa Barbara, U.S.A
htzheng@cs.ucsb.edu

Guang-Hua Yang
Dept. of EEE. The Univ. of Hong Kong
ghyang@eee.hku.hk

***Abstract*— Device coordination in open spectrum systems is a challenging problem, particularly since users experience varying spectrum availability over time and location. We propose a distributed coordination approach that handles spectrum heterogeneity without relying on the existence of a preassigned common control channel. Our approach carries potential to provide robust operation under network dynamics. While this approach can be implemented by upgrading the legacy protocol stack without modifying the MAC protocol, we also describe modifications to the MAC protocol that address spectrum heterogeneity and significantly improve system performance. Experimental results show that the proposed distributed coordination scheme outperforms the existing coordination schemes by 25-35% in throughput and provides 50% of delay reduction.**

## I. Introduction

To eliminate interference, historic (and current) spectrum policies allocate fixed spectrum slices to each wireless technology. Its centralized, static nature prevents users from dynamically reusing unused allocated spectrum, resulting in poor utilization and spectrum holes [4]. This motivates the *Open Spectrum* [1], [6] approach for spectrum access. While legacy license holders (*primary users*) have priority in spectrum access, unlicensed (*secondary*) users opportunistically utilize available spectrum without interfering with primary users. This results in efficient spectrum usage and simplified deployment for new applications.

In this paper we consider the coexistence of secondary users in open spectrum systems. In particular, efficient spectrum sharing among secondary users is integral to the success of open spectrum systems. Traditional approaches relying on a central server to observe and perform network-wide spectrum assignment is clearly inefficient for dynamic multi-hop networks. Instead, these networks require a decentralized access model, where users access spectrum based on locally observed availability. In this model, users must coordinate amongst themselves to optimize system performance [3] and exploit the benefit of open spectrum systems [2], [5], [11].

Effective and efficient coordination depends on fast dissemination of control traffic between neighboring users. Traditional coordination uses a common control channel known to all users [8], [9]. However, secondary users in open spectrum systems observe *spectrum heterogeneity*, spectrum availability that fluctuates over time and location. No common channels exist. One solution is to use an out-of-band licensed channel as the dedicated control channel for all users [10]. While simple, this approach has several drawbacks. First, it requires a static assignment of licensed spectrum before deployment, increasing complexity and cost. Second, the licensed channel's fixed bandwidth limits scalability in terms of device density, traffic and spectrum ranges. Finally, a simple jamming attack of the fixed control channel would disrupt the entire network.

In this paper, we propose an alternative *distributed coordination* scheme that addresses spectrum heterogeneity and the challenges above. Users in our approach self-organize into groups and coordinate using locally available common channels. This approach significantly improves scalability and reduces deployment costs.

This paper makes three contributions. First, we propose a group-based coordination scheme and develop distributed group setup and maintenance algorithms where users select coordination channels adaptively. Second, we provide two implementations of this scheme, one using existing device stacks with legacy MAC protocols and another using a new MAC protocol to explicitly address challenges from spectrum heterogeneity. Finally, we compare the proposed scheme to existing coordination schemes using extensive experiments.

The rest of the paper is organized as follows. We begin in Section II by describing spectrum heterogeneity and its impact on user coordination. Next, we present the general concept of distributed coordination and an algorithm to form coordination groups in Section III. In Section IV, we describe two implementations using legacy MAC protocols and a modified MAC protocol. We conduct experiments to compare our coordination scheme to existing common channel schemes in Section V. In Section VI, we discuss advantages and limitations of the work and outline future work. Finally, we conclude in Section VII.

## II. Spectrum Heterogeneity and Its Impact

In this section, we describe the challenges imposed on device coordination by spectrum heterogeneity. We first define the context and assumptions of open spectrum systems. We then show that while users do share significant spectrum with local neighbors, a common channel is rarely available to all users. Unless otherwise specified, we use "user" or "device" to refer to a secondary user.

### A. Assumptions and Network Model

We assume a network consisting of both primary users (license holders) and secondary users (unlicensed users). The spectrum is divided into non-overlapping orthogonal channels, which is the fundamental unit of spectrum usage. We note that

1-4244-0013-9/05/$20.00 ©2005 IEEE

each channel could be a physical channel, as in IEEE 802.11, or a logical channel associated with a spectrum region or radio technology. Spectrum licensed to primary users is accessible to secondary users if they do not interfere with primary users. We assume secondary users are static or quasi-static, and each user can accurately measure its available spectrum. While we assume secondary users can detect primary users, primary user detection is an open problem whose details are beyond the scope of this paper.

We assume that secondary users can communicate by selecting the same channel. This is different from the coexistence problems where devices equipped with different radios cannot communicate [7]. We assume that each user uses a single interface half-duplex transceiver, and can only transmit or receive on one channel at a time. This assumption is consistent with the implementation of WLAN devices. This is also true when only one radio can be invoked by a multi-radio device. We also assume that each channel has similar throughput capacity. This is because that channel quality fluctuates due to fading, shadowing and environmental factors, making it impractical to collect instantaneous channel quality in real time. Hence, a reasonable approach is to assume all channels result in a similar average throughput in this respect. Transmission errors can be handled by physical layer encoding or retransmissions and the impact is taken into account by the throughput measure.

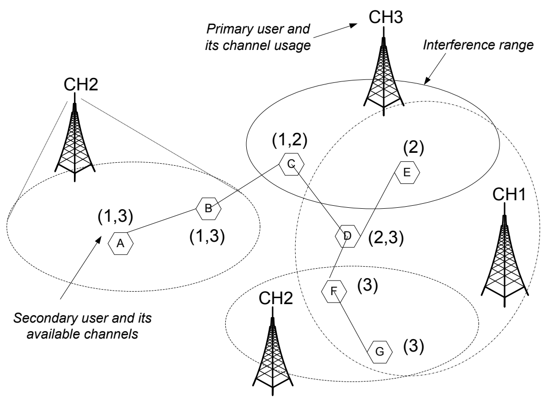


Fig. 1. An example open spectrum system

Based on these assumptions, we characterize spectrum usage using a simple binary interference metric. Assuming both primary and secondary users communicate using a predefined RF configuration (*i.e.* power, antenna), we determine interference condition by distance. That is, if the distance between a primary and a secondary user is less than $D_p$, channels used by the primary user are unavailable to the secondary user. Any two secondary users can communicate if they are within $D_c$ distance. Fig. 1 shows a sample network topology that consists of 4 primary users and 7 secondary users. The spectrum is divided into 3 channels and each primary user occupies one channel. An edge exists between any two secondary users within transmission range of each other.

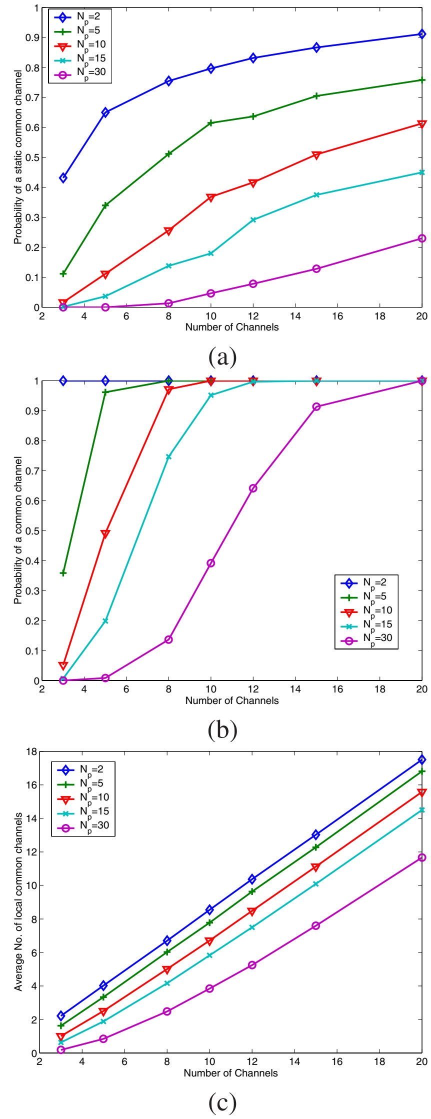


Fig. 2. Statistics of open spectrum systems. Assuming 40 secondary users, $D_p = 0.3$, $D_c = 0.15$ and 2-30 primary users. (a) The probability of the availability of a predefined common channel among all the users. (b) The probability of the availability of a common channel (not predefined) among all the users. (c) The average number of common channels each user shares with all of its neighbors.

## *B. Existence of A Common Channel*

In our example in Fig. 1, we see that neighboring users can share common channels without all users sharing a single common channel. To explore the availability of a common channel on a general topology, we randomly place 40 secondary users and vary the number of primary users on a 1x1 area. Primary users randomly select one channel to use, while we vary the number of channels between 3 and 20. We calculate the probability of a predefined channel being available to all users, and plot the average result of over 2000 topologies in Fig. 2(a). Our results using parameters $D_p = 0.3$ and $D_c = 0.15$ are representative of other configurations. We see that a commonly available channel is not guaranteed even for a small number of primary users. Such heterogeneity is a result of non-uniformed distribution of primary users' location and spectrum usage.

In Fig. 2(b), we examine the probability that any single channel is commonly available. For each topology, different channels become commonly available, and the system faces the challenge of reconfiguring the control channel as network topology changes. Even in this case, a commonly available channel is only guaranteed when the number of channels is large. Therefore, the common coordination channel approach does not apply to most open spectrum networks. Finally, Fig. 2(c) shows that each user shares a significant number of common channels with all of its neighbors. In other words, nearby nodes have very similar views of spectrum availability. We show in the next section how we can utilize these common channels to achieve efficient coordination.

## III. Distributed Coordination

As shown in Section II, spectrum heterogeneity reduces the feasibility of using a single common control channel. The fact that nearby users share similar spectrum availability motivates us to propose a distributed coordination scheme. In this section, we present the general concept behind our scheme and a detailed algorithm to form coordination groups.

### A. General Concept

In our distributed coordination scheme, users self-organize into local coordination groups based on similarity of available channels. Members of each group form a mini multi-hop network and use a common coordination channel. Assuming users maintain connections with each other through coordination, only members in the same group can directly communicate with each other.

Network connectivity is maintained by users at group boundaries that subscribe to multiple coordination channels. These "bridge" nodes relay traffic between groups and connect users in spatial regions with dissimilar spectrum availabilities. Fig. 3 sketches an example of coordination groups in an open spectrum system. In this example, node A, B and C form a coordination group using channel 1 as its coordination channel, C, D and E form another coordination group using channel 2, while D, F and G form the third group using channel 3.

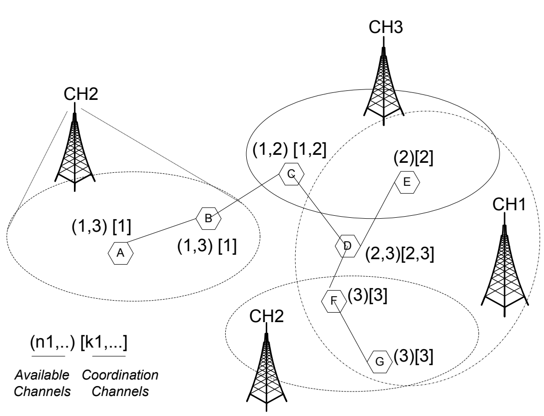


Fig. 3. An example coordination channel selection in open spectrum systems

In addition to providing a low-cost in-band coordination path for users with heterogeneous spectrum availability, this scheme has additional advantages over the common channel approach. These advantages also apply to networks with homogeneous spectrum availability.

*Scalability* – By organizing users into groups, coordination messages are distributed onto multiple coordination channels. This can prevent disruptions due to coordination traffic congestion. However, users can only monitor traffic and directly communicate with the members of groups they belong to.

*Robustness against jamming* – An attacker can easily disrupt a system using a fixed control channel by jamming it. Users in a distributed coordination system can find usable control channels to coordinate with neighbors unless all available channels are jammed. The distributed coordination approach can also be extended to impose security patterns to avoid disruption of coordination within each group. This will be addressed in a future study.

Each secondary device implementing distributed coordination must run two modules:

*Group Setup and Maintenance Module*: Each user executes a distributed algorithm to select coordination channels and form groups. The objective is to use a minimum number of channels to connect to all of its reachable neighbors. Two neighbors "connect" if they use the same coordination channel.

*Coordination Procedures Module*: Users in each group coordinate to exploit spectrum diversity: distributing transmissions across channels to avoid interference and improve throughput. The lifetime of a coordination channel is much longer than the duration of a data channel assigned to a device pair. The coordination procedures, implemented as MAC protocols, define user actions.

Next, we will describe the distributed coordination group setup algorithm. The detailed coordination procedures will be discussed in Section IV.

### B. Coordination Group Setup and Maintenance

To set up coordination groups, users need to obtain information about neighbors, particularly their spectrum availability. This is done through neighbor discovery, *i.e.* channel scanning and beacon broadcast. Note that since users need to broadcast beacons rotating through available channels, the neighbor discovery time is in general longer than that of a single channel network. After neighbor discovery, each device has a list of its neighbors, their available channels, and a schedule of time and channel to connect to each of them. This allows the user to send messages to all of its neighbors.

Based on these information, users execute *Algorithm 1* (in appendix) to select coordination channels. The procedure is a recursive distributed voting process where each user votes for a channel that provides the largest connectivity – the number of neighbors sharing the same channel. The vote is labelled with the value of the connectivity. The channel with the highest label in a neighborhood will be selected by all users in the neighborhood as the coordination channel. The process repeats until all users connect to their neighbors. Using the topology in Section II, Fig. 3 shows the result of coordination channel selection.

During network initialization, users execute Algorithm 1 to form coordination groups for the new topology. A user, upon joining an existing network, selectively joins existing coordination groups to connect to its neighbors. By eavesdropping on coordination messages from "bridge" users, mobile users obtain a list of coordination groups nearby and quickly subscribe to appropriate channels. This greedy algorithm produces minimum disturbance to existing users, however at the cost of sub-optimality in the coordination overhead. Note that the coordination overhead for each user is proportional to the number of coordination channels it subscribes to. Hence, for quasi-static networks, users should periodically perform network-wide group reconfiguration to reduce coordination overhead.

Coordination group formation also needs to adapt to primary users' spectrum activity. When a primary user starts to occupy a coordination channel, affected (secondary) users need to exit quickly from the channel and set up a different group. To avoid excessive setup time, users can maintain a backup coordination channel. The backup can be selected by executing Algorithm 1 again while removing the primary coordination channel from the pool of qualified channels. Before switching to the backup channel, users broadcast the decision on the current channel to inform neighbors about their coordination group changes. The broadcast duration should be kept to minimum to avoid excessive disturbance to the primary users.

## IV. Implementation on Ad hoc Networks

The concept of distributed coordination is applicable to a wide range of networks. In this section, we present two implementations on IEEE 802.11 ad hoc mode devices. In ad hoc mode, users can communicate with each other without any access point. The first implementation is through upgrading a legacy stack without modifying the associated MAC protocol, while the second implementation modifies the MAC protocol to address spectrum heterogeneity.

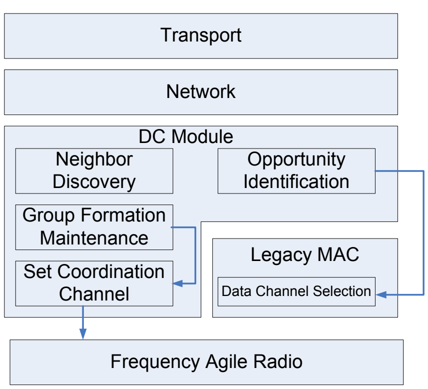


Fig. 4. Implementation with legacy stack

### A. Implementation using Legacy MAC Protocols

After forming a coordination group, group members execute a legacy MAC protocol to negotiate format of data transmissions. In this case, the proposed distributed coordination can be implemented by upgrading a legacy stack. Fig. 4 sketches the protocol stack, consisting of a distributed coordination (DC) module and a legacy MAC module.

The DC module performs neighbor discovery, interacts with physical layer to detect primary users, and identifies spectrum availability. It executes Algorithm 1 to form and maintain coordination groups, and configures the physical layer to switch to the proper coordination channel. Upon detecting primary users, the DC module broadcasts an alarm on coordination channels to neighbors within k-hops (*e.g.* k=2) to improve the speed of primary user detection for these neighbors. The MAC module interacts with each other to disseminate information of spectrum usage, and negotiate channels to transmit data packets. "Minor" modifications at the MAC module are required to restrict channel selection to the available spectrum.

This implementation allows realization of distributed coordination using legacy MAC protocols. However, since these MAC protocols are designed for homogeneous spectrum availability, this implementation faces a few challenges due to spectrum heterogeneity. First, "bridge" nodes need to switch between multiple groups to communicate with neighbors in different groups. The switch needs to be scheduled carefully to avoid inefficiency, especially when neighbors carry different volumes of traffic. Second, spectrum heterogeneity differentiates channels in terms of connectivity (*i.e.* the number of neighbors it allows to connect). Legacy MAC protocols ignore connectivity, potentially degrading communication efficiency. For example, after receiving a request from a neighbor, a user $n$ selects a data channel $m$ for a period of time. This decision affects the results of subsequent requests to $n$ from other neighbors – only requests from neighbors with channel $m$ available will succeed. Hence, channel selection needs to consider connectivity to allow efficient traffic multiplexing.

### B. Implementation using a New MAC Protocol

In this section, we show that some legacy MAC protocols can be modified to handle spectrum heterogeneity. To illustrate the modifications, we use the MAC protocol developed for IEEE 802.11 devices with multi-channel and single interface [8]. We call the modified MAC protocol heterogeneous distributed MAC (HD-MAC).

The legacy MAC protocol [8] divides transmissions into super-frames, each consisting of a beacon broadcast (BEACON), a coordination window (CHWIN) and a data transmission period (DATA). The single interface limits the device to accessing one channel at a time. Hence, instead of using a stand-alone control channel [9], the protocol uses a dedicated control window CHWIN to disseminate coordination information. During CHWIN, users switch to the common control channel (in our case, the coordination channel) to solicit transmissions and negotiate the channel to use. The coordination messages are sent during CHWIN following the CSMA/CA protocol. Each user records the number of successful negotiations on each channel by eavesdropping on coordination messages, and selects the channel with the

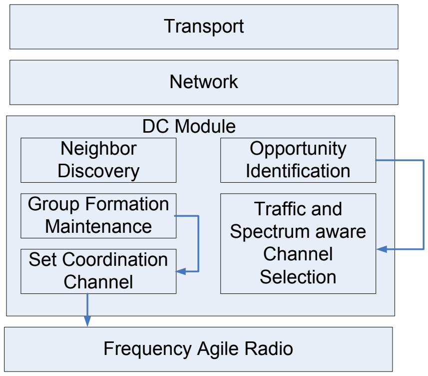


Fig. 5. Implementation with HD-MAC

minimum number of requests. At the beginning of DATA, users switch to the selected data channel to send packets. Such a frame-based MAC protocol requires tight synchronization which has been addressed in [8] for homogeneous spectrum availability.

To optimize performance, we make three modifications to the legacy MAC. First, we modify the CHWIN structure to allow "bridge" nodes to access multiple coordination groups in each super-frame. Second, we modify queue structure to a per-neighbor queue to avoid head-of-line blocking, and propose a peer_info structure to track neighbor information. Third, we propose a new data channel selection metric to jointly consider interference, connectivity and traffic load.

*1) CHWIN Structure*

Each user broadcasts a beacon signal during the BEACON period, rotating among its available channels in subsequent super-frames. That is, if a user has channel 1 and 2 available for transmissions, it attempts to broadcast beacon during even super-frames on channel 1 and odd super-frames on channel 2. This is referred to as *global beacon broadcast*. This is followed by a beacon broadcast on the coordination channel, referred to as *group beacon broadcast*. While group beacons provide quick neighbor discovery by transmitting beacon persistently from a channel, global beacons ensure discovery of any new user (when the coordination channel is not available to the new user). Users broadcast beacons according to IEEE 802.11 beacon procedures and the detailed procedure is omitted due to space limit.

During CHWIN, users with single coordination channel switch to the channel to coordinate. The CHWIN for "bridge" users is segmented into multiple slots, one for each coordination channel. The sequential access to coordination groups is mainly due to the restriction of single-interface device. One critical issue is the mapping between channels and slots. In particular, two "bridge" nodes within transmission distance can only coordinate if their mutual coordination channels map to the same slot.

One simple solution is to divide CHWIN into $M$ slots where $M$ is the total number of channels in the system, and preassign one channel to a slot. This is obviously inefficient when $M$ is large. We propose a hash compression scheme to divide CHWIN into $K \leq M$ slots ($K$ prefixed), and use a deterministic hash table to map $M$ channels to $K$ slots. This requires modifications to Algorithm 1: after selecting a coordination channel, each user removes all the channels mapping to the same slot from its candidate channel list. Since the mapping is many to one, users' connectivity might be degraded when $K << M$. Hence, the choice of $K$ and the hash table should be carefully planned. A simple hash table is modular-K, *e.g.* when $K = 2$, mapping evenly indexed channels to slot 0 and odd channels to slot 1. We also propose to rotate slot order in subsequent super-frames to allow fair access to multiple coordination groups. For example, when $K = 2$, in super-frame 1, the CHWIN consists of slot 0 and 1, and in super-frame 2, it consists of slots 1 and 0.

Fig. 6 shows the detailed operation of the legacy MAC implementation and the modified implementation. We see that with the legacy MAC, each "bridge" node can only access one coordination group during one super-frame, which is clearly inefficient. The modified MAC provides access to multiple coordination groups for a slightly shorter coordination duration. Note that the modified MAC requires devices to switch between channels much faster than the duration of the CHWIN slot.

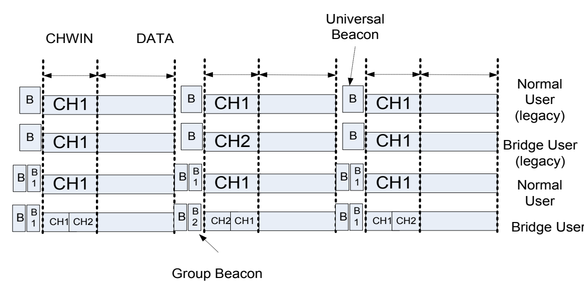


Fig. 6. MAC operation time line

*2) Per-Neighbor Queue and Peer Information*

In HD-MAC, each user tracks neighbors' spectrum and traffic information by eavesdropping on coordination messages and beacon broadcasts. The information is maintained by a peer_info entry for each neighbor. Spectrum heterogeneity also requires modifications to the queue structure. The legacy MAC protocols in general keep a single FIFO queue to accumulate traffic for all the neighbors. In open spectrum systems, it is possible that the current channel selected to send data packets is not available to the neighbor whose packets are at the head of the queue. To avoid head of line blocking, we propose that each user employs a per-neighbor queue structure that assigns one FIFO queue for each neighbor. During CHWIN, users initiate transmission requests to neighbors of the coordination group in a round-robin manner. During DATA, each user sends data packets in a round-robin manner to all the neighbors who have successfully negotiated the data channel. Algorithm 2 and 3 (in appendix) outline the pseudo-code on generating transmission requests and selecting the right neighbor to communicate with.

*3) Traffic and Connectivity Aware Channel Selection*

During each CHWIN slot, members of the corresponding coordination groups negotiate the data channel. While the legacy MAC protocol only considers neighbor interference, we propose a new channel selection metric that jointly considers traffic volume, connectivity and interference. A general user $u$, maintains a score for each channel $c$ as

$$\omega_u(c) = \lambda_{in} Q_{in}(c) + \lambda_{out} Q_{out}(c) - \lambda_f Q_f(c). \quad (1)$$

$Q_{in}(c)$ represents the estimated volume of incoming traffic that can be carried over channel $c$; $Q_{out}(c)$ is the estimated volume of outgoing traffic to its neighbors that can be carried over channel $c$; and $Q_f(c)$ is the estimated volume of traffic that could interfere with the user if all of them using channel $c$. $\lambda_{in}$, $\lambda_{out}$, $\lambda_f$ represents the relative weight of each traffic type.

At the beginning of CHWIN, each user updates $\{Q_{in}, Q_{out}, Q_f\}$. $Q_{out}(.)$ can be directly obtained from the current outgoing queue length, while $Q_{in}(.)$ and $Q_f(.)$ require knowledge of neighbors' queue status. In HD-MAC, queue status is piggybacked in coordination messages. By eavesdropping on these messages, users can progressively obtain accurate estimation of $Q_{in}(.)$ and $Q_f(.)$. The detailed procedure is shown as Algorithm 4 in the Appendix.

We also introduce an "outstanding" flag on channels. After a node pair selects a data channel, they set the channel to "outstanding". Their subsequent negotiations with other neighbors during the current CHWIN are based on the "outstanding" channel only. The "outstanding" flag can also be used to pre-select channels that can carry traffic with urgent QoS requirements. For example, a transmitter facing a threat of user queue overflow can set the channels that can carry the traffic to "outstanding", and will only choose data channel from the pool of "outstanding" channels.

To negotiate channel usage, each user $u$ maintains a candidate channel information $\Theta_u = \{c, \omega_u(c)\}_{c \in L(u)}$ on its available channels $L(u)$, and constantly updates $\Theta_u$ during CHWIN. If a user $u$ has "outstanding" channels, it will only include the "outstanding" channels in $\Theta_u$. The detailed negotiation procedure is shown below, and Fig. 7 plots the time line of messages.

(I). User $u$ selects a neighbor $v$ to send data packets. $u$ sends a CHRTS (channel request) message including its queue size (related to $v$) and $\Theta_u$ to $v$.

(II). Upon receiving a CHRTS for itself, user $v$ combines the channel score of both users. $v$ select a channel common to both $u$ and $v$ that maximizes $\min\{\omega_u(c), \omega_v(c)\}$, and sends a CHRES including the selected channel and the volume of pending packets. If there is no feasible channel, it sends a failure notice.

(III). Upon receiving a CHRES for itself, user $u$ sends a CHCFM to $v$ including the selected data channel and the length of pending packets, so that other neighbors can obtain queue information. User $u$ flags the selected channel as "outstanding" and updates $\Theta_u$ to only include the selected channel.

(IV). Upon receiving a CHCFM, user $v$ flags the selected channel as "outstanding" and updates $\Theta_v$ accordingly.

(V). User $x$ receives a CHRES or CHCFM not destined for itself, extracts queue status to update traffic information and channel score according to Algorithm 4.

To prevent communication disruption due to failure of receiving CHCFM, we propose to impose a guard-band between CHWIN and DATA. During guard-band, no CHRTS or CHRES should be sent.

## V. Experimental Results

In this section, we conduct experimental simulations to evaluate the performance of spectrum access through distributed coordination. We compare the performance of the distributed coordination approach to that of the dedicated licensed channel approach. We also compare the performance of implementations using the legacy MAC protocol and the modified MAC protocol.

We implement the coordination group formation in C/C++. We randomly deploy a large number of primary users and secondary users in a given area, and examine the effectiveness of distributed coordination under different network configurations. We also implement the spectrum access system with coordination and MAC protocols on NS-2 with CMU wireless extensions. This allows a detailed study of coordination efficiency under real TCP and UDP traffics. We simulate both CBR traffic and exponential on/off traffic, and similar conclusions are drawn. Each super-frame is 100ms and each CHWIN is of 24ms, further divided into 2 slot of 12ms each. The hash algorithm is the modular-2 operation.

### A. Connectivity of Distributed Coordination

We simulate a quasi-static ad hoc network by randomly placing primary and secondary users on $1 \times 1$ area, with $D_p = 0.3$ and $D_c = 0.15$. We examine the connectivity of distributed coordination, *i.e.* the percentage of neighbors (who share similar channels) connected under different value of $K$, the number of slots in each CHWIN. The hash function is modular-K. $K$ also represents the maximum number of coordination channels each user is allowed to use. Fig. 8 shows the average connectivity and the outage connectivity (90% of users have connectivity larger than the value), over 1000 random deployment of 40 secondary users and 40 primary users in the given area. Note that using a out-of-band dedicated coordination channel, the connectivity is 1. We see that the proposed distributed coordination with $K \geq 2$ provides similar connectivity compared to the dedicated channel approach.

We observe that for small number of channels the connectivity is close to 1. This is mainly due to that each secondary user has very small number of channels to use, so that the number of neighbors is close to zero. For these users, we set their connectivity to one. We also notice that there is a small dip in the average connectivity when the number of channels is 10. This is due to the assumption that each primary user randomly selects a channel to use. Increasing the number of channels initially increases the span of primary users' channel usage, leading to higher degree of spectrum heterogeneity at

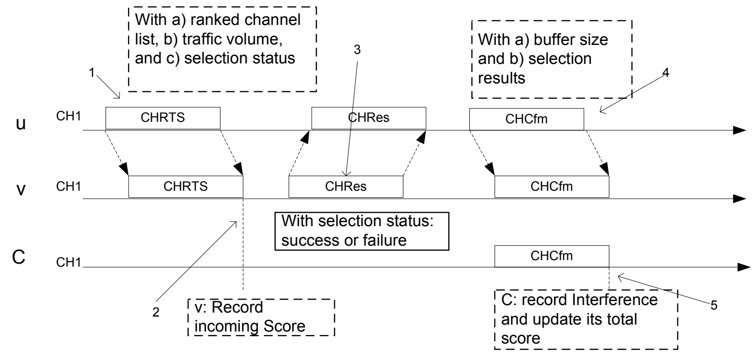


Fig. 7. Channel selection process

secondary users, and degrading the average connectivity. When the number of channels is sufficiently large, the degree of spectrum heterogeneity starts to decrease since the number of channels occupied by primary users in a local area is only a small fraction. As a result, the average connectivity starts to increase. However, there are still secondary users who are surrounded by primary users with diversified spectrum usage. Hence, the outage connectivity benefits less from introducing additional channels.

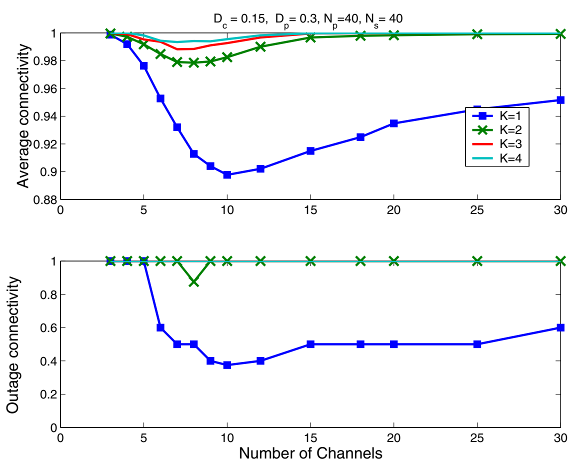


Fig. 8. The ratio of average and outage connectivity corresponding to hash table size $K$. The results are based on 1000 random deployment of 40 secondary users, 40 primary users in a $1 \times 1$ area with $D_p = 0.3$ and $D_c = 0.15$.

## *B. Comparison to Existing Coordination Schemes*

Using NS-2 simulations, we compare the performance of different coordination schemes in terms of user throughput. We examine the performance of out-of-band dedicated channel based scheme and the proposed distributed coordination scheme. For the dedicated channel based scheme, users switch to the dedicated control channel (*i.e.* channel 0) during CHWIN to negotiate transmissions. We consider a multi-hop scenario with two flows (Fig. 9). Using distributed coordination, there are three coordination groups using channel 1 to 3 respectively, mapping to $K = 2$ slots. Fig. 10 illustrates the performance using TCP and UDP flows. The proposed approach performs similar to the licensed channel approach.

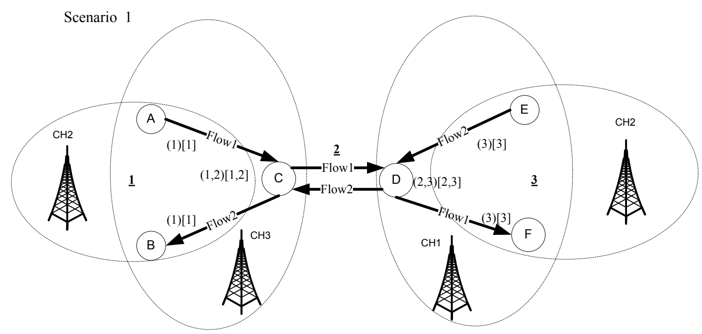


Fig. 9. Scenario 1 - the network topology and traffic flow used to compare the proposed distributed coordination with the extra licensed band scheme. Secondary users A, B and C are in coordination group 1 using channel 1, C and D are in coordination group 2 using channel 2, D, E and F are in coordination group 3 using channel 3.

In the example above, the coordination traffic is light. Next, we demonstrate the robustness of the distributed coordination against control traffic congestion. This shows that even for homogeneous spectrum availability, the proposed scheme outperforms the common channel based schemes. We setup a dense network with 52 users, each has 13 available channels. There are 26 FTP flows running between different user pairs. Using distributed coordination, some users exit from congested coordination groups to form new coordination groups. Results show that the distributed coordination prevents control congestion and achieves 29% improvement in throughput (see Table I).

TABLE I
FTP THROUGHPUT COMPARISON

| Scheme | Throughput(Mbps) | Comparison |
|---|---|---|
| Dedicated channel | 15.23 | - |
| Distributed Coordination | 19.65 | 29% |

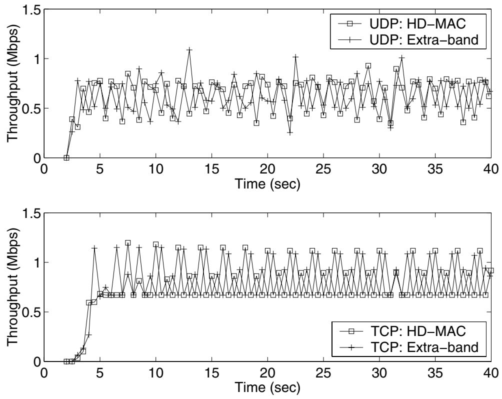


Fig. 10. Throughput comparison between the distributed coordination and the extra licensed band scheme.

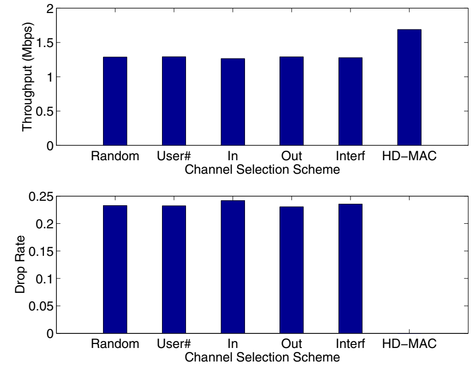


Fig. 12. Throughput and packet drop rate comparison of different channel selection metric, assuming Scenario 2.

## C. Comparison of MAC Implementations

Next we compare the performance of the legacy MAC protocol and the HD-MAC protocol, focusing on the channel selection metric. In the legacy MAC protocol, channel selection is based on the number of coordination requests on each channel (referred to as user #). To demonstrate the effectiveness of the proposed metric, we also include the performance of using just $Q_{in}$, $Q_{out}$, or $Q_f$, referred to as `in`, `out` and `interf` in the results. We also include a random selection, referred to as `random`. We set $\lambda_{in} = 0.3, \lambda_{out} = 0.5, \lambda_f = 0.2$, assigning outgoing traffic a higher priority to avoid buffer overflow.

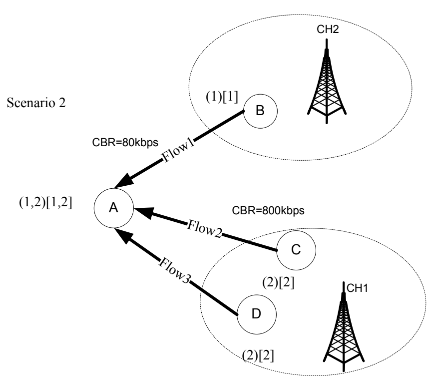


Fig. 11. Scenario 2: a single hop based network topology and traffic flow used to compare the proposed MAC implementation with the legacy MAC protocol. Secondary user A and B form coordination group 1 using channel 1, while A, C and D form coordination group 2 using channel 2.

We begin with a simple scenario where interference is not a concern. In Fig. 11, four users form 2 groups using channel 1 and 2 respectively. Users B, C, and D send CBR flows using UDP protocol to A, while flows from C and D have much larger volume. Fig. 12 shows that the proposed metric performs the best by jointly considering spectrum and traffic heterogeneity. In particular, user A assigns channel 2 a higher priority, and thus offers more transmission opportunities to user C and D. This significantly reduces the packet drop rate due to buffer overflow at C and D. It should be noted that the proposed HD-MAC protocol is different from traditional flow control schemes since it takes spectrum heterogeneity into account.

Next, we consider a multi-hop network with parallel traffic flows in Fig. 13. Users form two coordination groups using channel 2 and 3 respectively. User 1 and 2 send 80kbps flows to user 8 while users 3–7 send 120kbps flows to user 8. User 0 relays these traffic to user 8. Other parallel flows are of 40kbps. Fig. 14 illustrates the flow throughput and the packet drop rate. We observe that HD-MAC outperforms other approaches significantly: 35% throughput gain and nearly zero drop rate compared to 25% drop rate of the other schemes. For the other schemes, packet drops happen mostly at the "bridge" nodes and their neighbors, mainly due to improper channel selection that ignores connectivity. Fig. 15 compares the packet delay of different flows (average of flow 1 and 2, average of flow 3-7, average delay of flow 8-10) of different metrics, where HD-MAC leads to 50% reduction in packet delay.

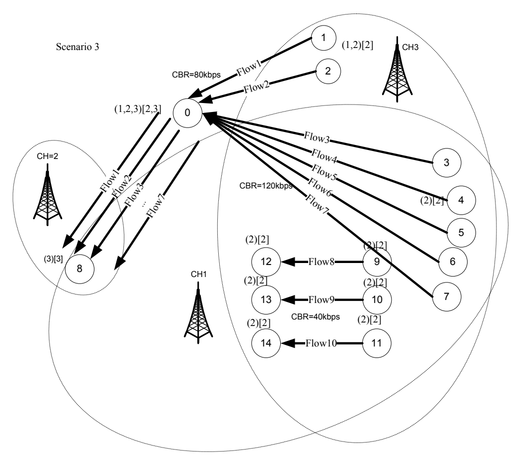


Fig. 13. Scenario 3: a multi-hop network topology and traffic flow used to compare the proposed MAC implementation with the legacy MAC protocol. Secondary user 0-7, 9-14 form coordination group 1 using channel 2, while user 0 and 8 form coordination group 2 using channel 3. Secondary user 0 is a bridging node who also forwards packets from user 1-7 to user 8.

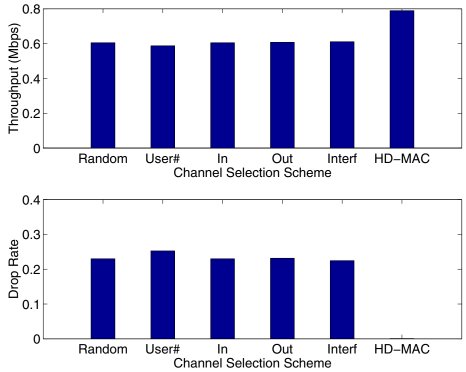


Fig. 14. Throughput and packet drop rate comparison of channel selection metric, assuming Scenario 3.

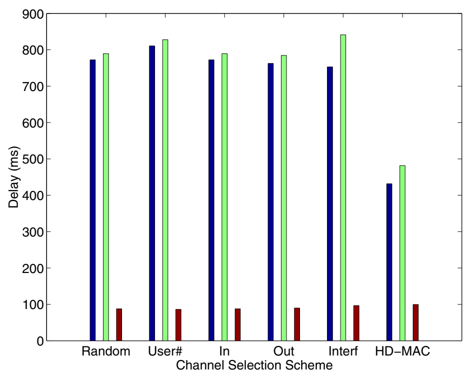


Fig. 15. Packet delay comparison of channel selection metric, assuming Scenario 3. The bars corresponding to each selection metric represent the packet delay of flow 1-2, 3-7 and 8-10 respectively.

## VI. Discussions

In this paper, we focus on efficient spectrum sharing through distributed coordination. In addition to allow negotiations of spectrum usage, the proposed coordination scheme provides additional functionalities that can be used in dynamic spectrum allocation networks. In particular, coordination channels provide a fast and efficient way to broadcast control messages to neighbors, which is essential for users in multi-hop networks to perform route discovery and maintenance. The proposed framework also promotes development of powerful protocols to exploit group collaboration. For example, secondary users in the same coordination group can share the task of spectrum sensing and primary user detection, and broadcast spectrum status to neighbors. This can significantly reduce power consumption and avoid communication disruption.

We note that a successful implementation of the proposed system requires tight synchronization among secondary users. The proposed time frame based transmission structure requires users in the same coordination group to switch to the coordination protocol within a short period of time. This implies additional complexity on hardware design - the channel switch delay should be much shorter than the length of CHWIN slot.

In this paper, we focus on the task of spectrum/channel usage coordination which is in general a responsibility of MAC layer. We are aware that modifications are required at higher layers to respond to dynamically changing spectrum availability among secondary users. In particular, route selection impacts the traffic load on each link and the amount of spectrum/bandwidth required. Joint selection of route and spectrum could make better usage of spectrum and improve end-to-end performance for multi-hop transmissions. Group based coordination allows collaborations and fast information dissemination among users. In particular, it provides a simple and energy efficient procedure to broadcast route discovery message. We are currently investigating a spectrum aware routing protocol that adapts route selection to spectrum fluctuations.

## VII. Conclusion

We present a distributed coordination scheme to explore under-utilized spectrum in open spectrum ad hoc networks while addressing spectrum heterogeneity. Users dynamically select the coordination channel based on local conditions, eliminating the need of a common coordination channel. The proposed approach can be implemented using existing device stacks with legacy MAC protocols or using a new MAC protocol to explicitly address challenges from spectrum heterogeneity. Experimental results show that our approach significantly outperforms existing coordination schemes.

## References


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

[2] Cao, L., and Zheng, H. Spectrum allocation in ad hoc networks via local bargaining. In *IEEE SECON* (Santa Clara, CA, Sept. 2005).

[3] Ephremides, A., and Truong, T. Scheduling broadcasts in multihop radio networks. *IEEE Trans. on Communications 38*, 4 (April 1990), 456–460.

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

[5] Peng, C., Zheng, H., and Zhao, B. Y. Utilization and fairness in spectrum assignment for opportunistic spectrum access. *Mobile Networks and Applications (MONET)* (2006). to appear.

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

[7] Raychaudhuri, D., and Jing, X. A spectrum etiquette protocol for efficient coordination of radio devices in unlicensed bands. In *Proc. of IEEE PIMRC* (Sept. 2003).

[8] 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.

[9] 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.

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

[11] Zheng, H., and Peng, C. Collaboration and fairness in opportunistic spectrum access. In *Proc. 40th annual IEEE International Conference on Communications* (June 2005).


## Appendix

### A. Algorithm 1: Distributed coordination channel selection

1: **Init**– For each new user $n$, Let $O(n)$ represent the set of available channels; Let $\chi(n)$ be the set of neighbors of $n$; Let $\chi\kappa(n)$ represent the set of neighbors that n has not connect to; Let $C(i)$ represent the coordination channel(s) that user $i$ has selected.

2: Set $\chi\kappa(n) = \chi(n)$, $C(n) = \phi$.
3: **for** all $i, \{i \in \chi\kappa(n)\ \&\ C(i) \neq \phi\}$, **do**
4: **if** $C(i) \cap O(n) \neq \phi$

$$C(n) \longleftarrow C(n) \cup \{C(i) \cap O(n)\}$$
$$\chi\kappa(n) \longleftarrow \chi\kappa(n) \setminus i.$$

5: **end if**
6: **end for**
7: **if** $\chi\kappa(n) \equiv \phi$, **then**
8: end
9: **else** $\{$
10: Define a connectivity level of a channel $m$

$$T(m,n) = \sum_{i \in \chi\kappa(n)} 1(m \in O(i));$$

11: User $n$ constructs and broadcasts message $REQ_n$ as the triplet $\{n, m_n, L(n)\}$ where

$$m_n = \max_m T(m,n)$$
$$L_n = T(m_n,n) + Rand(0.5,1)\},\ if\ m_n \in C(n)$$
$$= T(m_n,n) + Rand(0,0.5)\},\ if\ m_n \notin C(n).$$

12: After $\delta$ time, user $n$ collects broadcast from non-connected neighbors and select the broadcast with the highest connectivity level as the coordination channel, *i.e.*

$$m^* = \underbrace{argmax}_{i \in \chi\kappa(n),\ m_i \in O(n)} L(i),$$
$$C(n) \longleftarrow C(n) \cup m^*.$$

13: User $n$ broadcasts a message $SET_n = \{n, C(n)\}$.
14: After $\delta$ time, user $n$ collects broadcast from non-connected neighbors and update

$$\chi\kappa(n) \longleftarrow \chi\kappa(n) \setminus i, \quad if C(i) \cap C(n) \neq \phi.$$

15: If $\chi\kappa(n) \equiv \phi$, **End**, else **go to** line 9.
16: $\}$
17: **end if**

### B. Algorithm 2: Initiate channel coordination request

$chrtsSent \leftarrow 0$
$loopCount \leftarrow 0$
$i \leftarrow coTurn$
$N \leftarrow$ Neighbor Number
**while** $chrtsSent \equiv 0$ And $loopCount < N$ **do**
  $loopCount \leftarrow loopCount + 1$
  **if** this channel is $nQ_i$'s coordination channel **then**
    **if** $nQ_i$ has pending packet and $nQ_i$ is not negotiated **then**
      $chcoRequest(dst_i)$
      $chrtsSent \leftarrow 1$
      $nQ_i.negotiated \leftarrow true$
    **end if**
  **end if**
  $i \leftarrow (i+1)$ mod $N$
**end while**
$coTurn \leftarrow i$

Here, the $chcoRequest(dst_i)$ is the procedure that generates the CHRTS frame to destination $dst_i$ for neighbor queue $nQ_i$ and starts the related MAC timers (backoff timer of IEEE 802.11 DCF) if necessary.

### C. Algorithm 3 : get a qualified queue for transmission in data period

$pktDeqed \leftarrow 0$
$i \leftarrow dataNext$
**repeat**
  **while** $qHandler_i$ is invalid and we have not traversed the neighbor queues **do**
    $i \leftarrow (i+1)$ mod $N$
  **end while**
  **if** we have traversed but get nothing to do **then**
    break
  **end if**
  **if** $qHandler_i$ is valid and $nQ_i.selected \equiv true$ **then**
    call $qHandler_i$
    **if** $pktDeqed \equiv 1$ **then**
      $dataNext = (i+1)$ mod $N$
    **end if**
  **end if**
  $i \leftarrow (i+1)$ mod $N$
**until** $pktDeqed \equiv 1$ or We've traversed all the neighbor queues

Here the $qHandler_i$ is the queue handler for $nQ_i$ that is used to resume upper layer queues after MAC has finished current packet processing.

### D. Algorithm 4: update the score of each channel

**switch**(Event Type)$\{$
**case** Beginning of Coordination Period:
  $Q_{in}(t) = (1-\alpha)Q_{in}(t-1) + \alpha Q_{in}(t-1)_m$
  $Q_f(t) = (1-\alpha)Q_f(t-1)$
  $Q_{out}(t) = Q_{out}(t)_m$
  $\omega(t) = \lambda_{in}Q_{in}(t) + \lambda_{out}Q_{out}(t) - \lambda_f Q_f(t)$
  Mark "outstanding" channel if needed.
  **break**;
**case** Overhearing CHCfm:
  $Q_f(t) = Q_f(t) + \alpha Q_f(msg)$
  $\omega(t) = \omega(t) - \lambda_f \alpha Q_f(msg)$
  **if** we have not selected a channel **then**
    Mark "outstanding" channel if needed.
  **end if**
  **break**;
**case** Receiving CHRTS:
  Record $Q_{in}(t)_m$
  **break**;
$\}$

where $\omega(t)$ denotes the total score, $Q_{in}(t-1)_m$ denotes the incoming traffic measured at the end of last $t-1$ coordination period, and $Q_{out}(t)_m$ denotes the outgoing traffic volume measured just before the coordination period begins. $Q_f(msg)$ is the eavesdropped traffic volume on a given channel from CHCfm message.