This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Secon 2009 proceedings.

Cognitive DISH: Virtual Spectrum Sensing Meets Cooperation

Tie Luo and Mehul Motani
Department of Electrical and Computer Engineering
National University of Singapore
Email: {elelt, motani}@nus.edu.sg

***Abstract*—Cognitive radio technology increases spectrum utilization by enabling secondary users to opportunistically use the spectrum when primary users are inactive. Secondary users use *spectrum sensing* to detect the presence of primary users in order to avoid causing harmful interference. To the best of our knowledge, all existing spectrum sensing methods are essentially *physical spectrum sensing*, in the sense that nodes physically tune their radio to each frequency band to sense the spectrum. In this paper, we propose a complementary approach, *virtual spectrum sensing*, which achieves the same goal but only senses a very small portion of the spectrum. This approach enables a Distributed Information SHaring (DISH) mechanism, where neighboring users cooperatively share spectrum usage information (obtained from virtual spectrum sensing) with users who need it in decision making. This paper presents an application of DISH to cognitive radio networks. We provide a Cognitive DISH framework which describes guidelines for cognitive radio protocol design based on virtual spectrum sensing and DISH. Under this framework, we design a protocol, VISH-I, and evaluate its performance via simulations. As the number of secondary users increases, the interference caused to primary users results in only 5% performance degradation, but the overall channel utilization is increased by 87–203%. In addition, to demonstrate that virtual sensing is complementary to physical sensing, we design a *hybrid spectrum sensing* protocol, VISH-II, which improves performance by 7–50% over VISH-I.**

I. INTRODUCTION

In the past few decades, the convenience of anytime anywhere connectivity has significantly boosted the penetration of wireless communications and hiked up the demand for spectrum resources. However, studies surprisingly show that the spectrum utilization is low. For instance, the FCC reported that 70% of the allocated spectrum in the US is not utilized [1]. To exploit this opportunity, cognitive radio was proposed by Mitola [2] based on software defined radio. This has attracted attention from researchers and standardization bodies. For example, the IEEE 802.22 working group is defining wireless regional area networks (WRAN) as an application of cognitive radio, aiming to efficiently use VHF/UHF TV bands.

A cognitive radio network consists of primary users, who are licensed or paying users, and secondary users, who are unlicensed or non-paying users, and embodies two principles:

1) Primary users *need* not care about secondary users when using the spectrum.
2) Secondary users attempt to use the spectrum opportunistically, with the constraint that they should not cause harmful interference to primary users.

Usually, work in cognitive radio networks assumes that secondary users have very limited knowledge about primary systems, i.e., a primary system is viewed as legacy with unknown PHY or MAC layer characteristics. As such, a cognitive radio must be "intelligent" enough to figure out those characteristics (of PHY layer, especially) so that it can detect the presence of primary users to avoid interfering with them.

In this paper, we relax this assumption and allow richer interaction between primary and secondary systems. Specifically, we assume that secondary users are able to decode the control messages sent by primary users and know the semantics conveyed by the messages. This is feasible because many primary systems are based on standards and hence the PHY and MAC layer details are readily available. For example, Singapore has deployed an island-wide WiFi Internet access infrastructure with two access modes: Wireless@SG as a free service and Wireless Plus 3000 as a premium service [3]. This is also an example of paying and non-paying users. On the other hand, we do not assume that secondary users can decode data messages, because data payload is usually encrypted for security and privacy concerns. In any case, we ensure that the two principles of cognitive radio networks are strictly abided by.

Cognitive radio networks with this richer interaction would facilitate *spectrum sensing*, whose goal is to detect primary users or, alternatively, spectrum holes or white spaces. The conventional approach to spectrum sensing is to physically scan every possible spectrum band continually for the presence of primary users. Typical techniques include energy detection, matched filter detection [4], cyclostationary detection, wavelet detection, and cooperative spectrum sensing and data fusion [5]. Each of these techniques has advantages and disadvantages, and a trade-off between complexity and accuracy always has to be made carefully [6]. We refer to these techniques collectively as *physical spectrum sensing*.

In this paper, we propose a different but complementary approach, *virtual spectrum sensing*, to fulfill the same task. The basic idea is to "sense" the spectrum usage by sensing only a small portion of the spectrum (e.g., a control channel), where the collected information maps to the usage of the entire spectrum. Hence ultimately, a node virtually senses all of the frequency bands.

An important advantage over physical spectrum sensing is that virtual spectrum sensing not only detects the *presence*

978-1-4244-2908-0/09/$25.00 ©2009 IEEE

of primary users, but also captures information about the *activities* of primary users (e.g., who is communicating, on what channels, when and for how long). This enables a Distributed Information SHaring (DISH) mechanism, whereby idle neighboring nodes cooperatively share their acquired spectrum usage information with transmitter-receiver pairs to aid in their decision making. This notion of DISH was first proposed in [7, 8] and then explored in [9, 10] from different aspects. In cognitive radio networks, spectrum usage information is vital for opportunistic and efficient resource utilization, and hence it would be beneficial to incorporate DISH to enhance the effectiveness of virtual spectrum sensing.

This paper presents an application of DISH to cognitive radio networks. Specifically, the contributions of this paper are summarized below:

1) We introduce virtual spectrum sensing as a complementary approach to physical spectrum sensing for user detection. Virtual sensing allows for sensing the entire spectrum without physically tuning radios to each band, and yields richer information than physical sensing as it detects user activities in addition to presence.
2) We propose a Cognitive DISH framework by integrating DISH into virtual spectrum sensing. This framework provides protocol design guidelines for cognitive radio networks, with the objectives of (1) minimizing the interference caused by secondary users to primary users and (2) maximizing overall channel utilization.
3) We design a concrete Cognitive DISH protocol under the framework and evaluate its performance via extensive simulations. This protocol, VISH-I (Virtual spectrum sensing with Information SHaring), is shown to achieve the two objectives: it limits the harmful interference to less than 5% performance degradation for primary users, and increases the overall channel utilization by 87–203%.
4) As virtual sensing is complementary to physical sensing, we show that these two approaches can be combined by designing another protocol, VISH-II, based on *hybrid spectrum sensing*. Further sets of simulations show that, in addition to achieving the same objectives as VISH-I, it improves the performance of VISH-I by 7–50%.

The rest of this paper is organized as follows. Section II describes the Cognitive DISH framework. Section III and IV give the protocol design of VISH-I and VISH-II, which are based on virtual spectrum sensing and hybrid spectrum sensing, respectively. Their performance is then evaluated in Section V. Lastly, Section VI discusses related issues and Section VII gives concluding remarks.

II. Cognitive DISH Framework

A. Network Model

We consider a static or quasi-static cognitive radio network consisting of both primary users and secondary users. The network structure can be either flat like ad hoc networks or hierarchical like mesh networks, and communications can be either distributed or partially centralized.[1] Two nodes can communicate directly if they are within the transmission range of each other, and communicate via multi-hop otherwise. A small portion of the spectrum is used as the one common control channel, known to both primary and secondary users, for users to disseminate and collect spectrum usage information (i.e., virtual spectrum sensing) and share their collected information (i.e., DISH). The remaining major portion of the spectrum is divided into multiple data channels for users to perform actual data transmissions. Each user is equipped with a single half-duplex transceiver, which can dynamically switch between these channels but can only use one at a time.

B. Virtual Spectrum Sensing

Detecting the presence and/or activities of other users is not a requirement for secondary users only; every user needs to perform this for collision avoidance unless the network is fully centralized and resource allocation is completely static. In the Cognitive DISH framework, this user detection task is fulfilled using virtual spectrum sensing, whereby users learn spectrum usage by collecting the information disseminated in a small portion of the spectrum, rather than by physically scanning the entire spectrum or tuning radio onto each specific band of interest. This is made feasible by dividing the spectrum into multiple channels, indexing each of them, and broadcasting the indexes of the channels being used in a certain band.

Besides detecting busy bands, virtual spectrum sensing can also obtain other critical information such as how long the channel will be occupied and who are the occupiers. This opens a wide window of possibilities to explore, among which a DISH mechanism is incorporated into this framework to enhance the effectiveness of virtual spectrum sensing.

C. DISH

In traditional MAC protocols, nodes acquire control information and use it themselves. Such examples include the IEEE 802.11 [13], where nodes schedule data transmission based on RTS or CTS that they overhear, and many self-learning protocols [14, 15]. DISH, on the other hand, exploits control information to a further extent by allowing idle neighbors to *share* it with nodes who need it. This idea has been shown by prior studies [7]–[10] to be very effective and able to improve system performance substantially.

Specifically, when there are multiple channels available, DISH can be used to solve two problems: (1) channel conflict problem—a node selects a channel (or band) which is already being in use by another node in the proximity, and (2) deaf terminal problem—a node initiates communication with another node who is on a different channel. The cause to both problems is nodes' insufficient knowledge about channel usage, and this knowledge insufficiency can be compensated for via information sharing among nodes. In fact, DISH is the

[1] An example of centralized cognitive radio networks is the IEEE 802.22 WRAN, and two examples of distributed cognitive radio networks is cognitive ad hoc networks [11] (chapter 8) and xG networks [12] (the part without infrastructure).

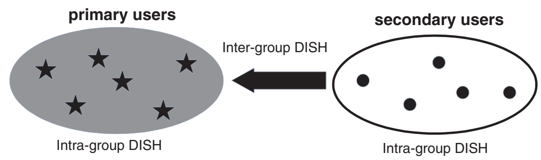


Figure 1: The two types of DISH.

distributed flavor of *control-plane cooperation*, as opposed to traditional data-plane cooperation which refers to intermediate nodes helping relay data for source-destination pairs.

In the Cognitive DISH framework, we classify two types of DISH: intra-group DISH and inter-group DISH, illustrated in Fig. 1. Intra-group DISH is a system design choice, i.e, whether secondary users help (i.e., provide cooperative information to) secondary users and whether primary users help primary users depend on the system design. On the other hand, inter-group DISH can be asymmetric, meaning that secondary users can help primary users (if they understand) but primary users need not help secondary users.[2] This conveys an important implication: in a Cognitive DISH network, secondary users are not merely opportunistic *consumers* of system resource as in existing cognitive radio networks, but can also *contribute* to the system by sharing critical control information with primary users for primary users' sake. This can be viewed as a price for secondary users to afford as unlicensed consumers, or a reward to primary users.

D. User Differentiation

It is essential to prioritize primary users and secondary users in accessing the spectrum as they basically share this resource. In principle, primary users are privileged users but secondary users can only have opportunistic access. Based on this, the following guidelines are specified:

1) *Control channel contention*: The control channel serves as a rendezvous for all users (primary and secondary) to set up communication, and thus contention will arise among users when accessing this channel. We require that, for every control channel contention between a primary user and a secondary user, the primary user must have much higher chances to win, or if possible, be guaranteed to win.
2) *Data channel occupancy*: We aim for primary users to achieve high throughput and low delay, and even when they do not inject sufficient traffic, we aim at maintaining a high overall channel utilization provided that secondary users have backlogged data. Accordingly, we allow a primary user to transmit on a data channel for as long as possible if only it does not monopolize the channel and starves other *primary* users. On the other hand, a secondary user should use a data channel vigilantly and release it quickly upon the return of a primary user. Nevertheless, when primary users are long-term inactive, secondary users should be given the chance to prolong their channel occupancy.

[2]Possible extensions of asymmetric DISH are discussed in Section VI-B.

III. Cognitive DISH Protocol

Under the framework described above, we design a protocol called VISH-I which can be used in multi-hop networks. For the ease of description, we first define two terms below:
A complete communication process between a transmitter and a receiver is composed of a *control session* and a *data session*. A control session is a process in which the transmitter and the receiver exchange control packets to establish a data session. A data session is a process in which the transmitter delivers data packets to the receiver.

In our protocol, control sessions are performed on the control channel and data sessions are performed on data channels.

A. Control Session

The design of control session adapts the handshake of CAM-MAC [8] with a different collision avoidance scheme and a different set of frame format. The handshake is shown in Fig. 2. A transmitter and a receiver first perform a PRA/PRB exchange like the IEEE 802.11 RTS/CTS handshake, where PRA carries a data channel index selected by the transmitter and PRB duplicates this channel index (if agreed by the receiver). This PRA/PRB conversation has a dual purpose: (1) to suppress neighboring nodes from generating interference, like what the RTS/CTS does, and (2) to probe neighboring nodes if there is a problem (channel conflict or deaf terminal) associated with this conversation.

If no problem is associated with the conversation, this transmitter-receiver pair proceeds to a CFA/CFB exchange, in order to confirm (to their respective neighbors) that the data session will actually start. After that, this pair of nodes switch to their agreed-on data channel.

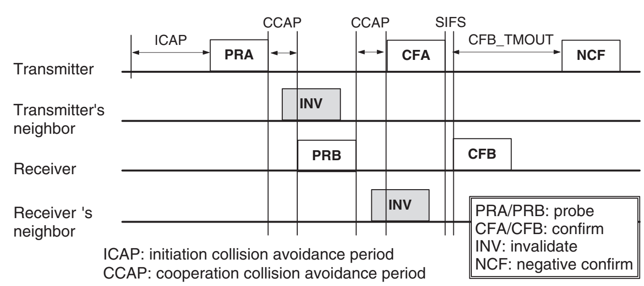


Figure 2: Control session.

1) DISH-based Cooperation: If there is a channel conflict or deaf terminal problem associated with the PRA/PRB conversation, DISH will be triggered. Due to virtual spectrum sensing, neighboring nodes listen to the control channel during idle periods,[3] so that they can collect data channel usage

[3]A node is considered idle if it is not transmitting or receiving for its own sake, i.e., it is idly waiting, backing off, or overhearing messages not addressed to it.

information broadcast from nearby transmitter-receiver pairs' control sessions. Therefore, when a problem occurs, some neighbors can identify the problem. In that case, each of them will attempt to send a cooperative message, INV, to inform the transmitter (on hearing PRA) or the receiver (on hearing PRB) to back off to terminate the control session. Note that the receiver can also send INV if it does not agree with the transmitter's channel selection, since it is also one of transmitter's neighbors.

However, sending cooperative messages brings forth two issues. The first is *cooperation coordination*, which is to resolve cooperative message collision caused by multiple neighbors identifying the same problem and sending cooperative messages simultaneously. We address this by inserting two cooperation collision avoidance periods (CCAP) into the control session, one between PRA and PRB and the other between PRB and CFA, as indicated in Fig. 2. Each node with INV to send will sense the control channel for $U[0, T_{ccap}]$ before sending INV, where T_{ccap} is the length of CCAP and $U[\cdot]$ stands for uniform distribution. If the node senses the channel becomes busy (due to some other node sending out INV), it will cancel its own INV. This mitigates 70–85% collisions in multi-hop scenarios as observed in our simulation. In the rest cases, we make those collisions *meaningful* by requiring nodes to only express *negative* response (via INV) and not to respond if they agree with the PRA/PRB conversation. Therefore, a collision in this scenario implies that at least one INV has been sent and hence the control session should still be terminated.

The second issue is *cooperation interference*, meaning that cooperative messages may cause interference to nearby ongoing control sessions that are *without* channel conflict or deaf terminal problems. We address this using a "loyal period" similar to the IEEE 802.11 NAV, where a node enters a loyal period when it overhears and agrees with a control session, and exit at the end of the control session. During this period, the node keeps silent even if it (1) identifies a problem associated with another nearby control session (say H) or (2) receives a PRA addressed to it (i.e., it is an intended receiver). This is reasonable because, for (1), session H can likely obtain cooperation from other neighbors, and for (2), this node can respond to a PRA retry later because a loyal period expires quickly.

2) Control Channel Contention: To initiate a control session, a transmitter will perform a clear channel assessment (CCA) on the control channel, which is a simple CSMA mechanism performed during the initiation collision avoidance period (ICAP) indicated in Fig. 2. To implement prioritized control channel contention (Section II-D) as prescribed by the framework, we set

$$\begin{aligned} T_{ICAP}^{pri} &\sim U[T_{gap}, T_{ctrl}], \\ T_{ICAP}^{sec} &\sim U[T_{ctrl}, 8T_{ctrl}], \end{aligned}$$

where T_{ICAP}^{pri} and T_{ICAP}^{sec} are the lengths of ICAP for primary and secondary users, respectively, T_{ctrl} is the duration of a successful control session, $T_{gap} = 2T_{ccap} + T_{prb}$, and T_{prb} is the transmission time of PRB. Note that T_{gap} is the longest quiescent period within a control session, during which there is no signal to detect for the transmitter's and not the receiver's neighbors. Therefore, as $T_{gap} < T_{ICAP}^{pri} < T_{ICAP}^{sec}$, an ongoing control session will not be interfered by a node who initiates a new control session right after switching to the control channel.

Algorithm 1 Linear increase aggressive decrease (LIAD)

$q^{sec} \leftarrow 1$
while TRUE **do**
    Timer $\leftarrow T_{idle}^{pri}$
    while Timer not expired **do**
        if Any activity of a primary user is detected **then**
            $q^{sec} \leftarrow 1$
            break {jump out of the inner loop}
        end if
    end while
    if Timer expired $\wedge$ $q^{sec} \leq Q^{sec}$ **then**
        q^{sec}++
    end if
end while

Revisiting the framework, this design guarantees a primary user to win *every* control channel contention over a secondary user if they start to contend simultaneously, which is obvious to see, and to win with much higher chances in random scenarios. Relevant performance will be evaluated in Section V.

B. Data Session

A data session starts immediately after a transmitter-receiver pair switches to the data channel negotiated during the control session. To prioritize *data channel occupancy* as specified in Section II-D, we design an adaptive and user-differentiated *packet train* mechanism illustrated in Fig. 3. Using a packet train allows a transmitter to send a burst of data packets consecutively to a receiver without performing a control session for each data packet, which amortizes the control packet overhead and mitigates the control channel bottleneck problem.[4]

Our packet train features a linear increase aggressive decrease (LIAD) algorithm, which dynamically adjusts the train size for secondary users. Define *train size quota* as the number of data packets that are allowed to transmit in a single batch (i.e., with only one control session). Denote the train size quota of a primary user by Q^{pri} and that of a secondary user by Q^{sec}, both of which are predefined constants and $Q^{sec} \leq Q^{pri}$. Each secondary user maintains an actual train size quota, q^{sec}, which varies within $[1, Q^{sec}]$ following a LIAD algorithm shown in Algorithm 1.

In this LIAD algorithm, T_{idle}^{pri} is a constant interval and if no activity of primary users is detected during this interval,

[4]Control channel bottleneck is a scenario where free data channels are available but can not be used because congested traffic on the control channel prevents nodes from setting up new communications.

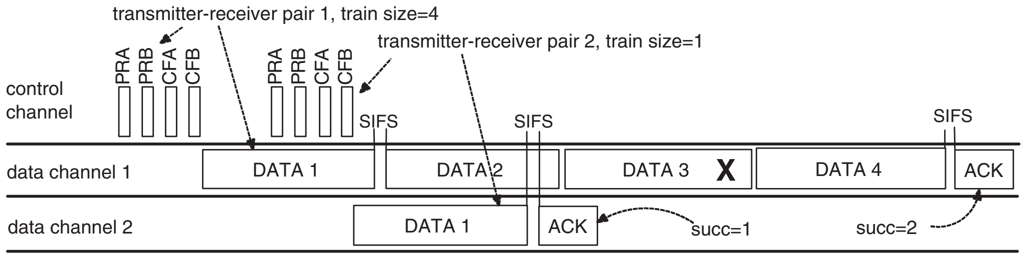


Figure 3: Data session of VISH-I. Transmitter-receiver pair 1 and 2 can be either primary or secondary users as train size depends on both train size quota and the actual packets in queue. `succ` is a field in `ACK` (cf. Fig. 4) indicating the number of successfully received DATA before the first failure.

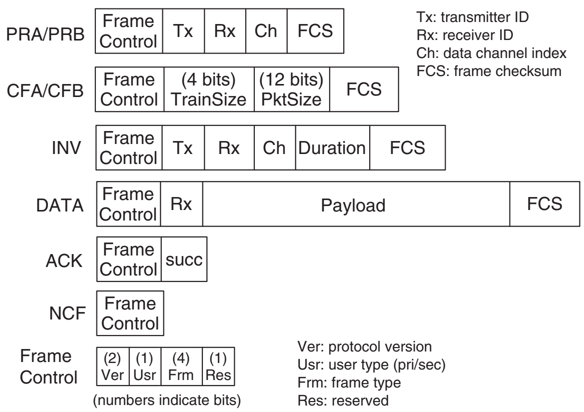


Figure 4: Frame format.

q^{sec} increases by 1. Once any activity from a primary user is detected, q^{sec} is reset to 1. This ensures that (1) secondary users vacate the spectrum promptly[5] when primary users become active, and (2) the spectrum does not suffer from long-term under-utilization in scenarios where primary users only inject sparse traffic into the network.

LIAD is a tailored version of additive increase multiplicative decrease (AIMD) algorithm [16] proposed for TCP congestion avoidance. The rationale of the aggressive decrease of q^{sec} (reset to 1) is that, when a primary user become active again after a fairly long idle time, this user will usually have a burst of (instead of just one) incoming data packets. Therefore, it would be more efficient to directly reset q^{sec} to the minimum instead of iterating several times of multiplicative decreases.

C. Frame Format

The frame format is shown in Fig. 4. User type (primary or secondary) is differentiated in the Frame Control field. The 4-bit `TrainSize` field in CFA/CFB allows for up to 15 packets in a train, and the 12-bit `PktSize` field allows for up to 4095 bytes in a data packet.[6] Hence together, a maximum of 61,425 bytes (4095×15) can be sent in one packet train.

[5]The IEEE 802.22 specifies 2 seconds for this "grace period".

[6]Variable-size data packets can be allowed via a simple modification: replacing the `TrainSize` and `PktSize` with a `TotalSize` (or `Duration`) field, and adding a `Size` field into each DATA.

Table I: Channel Usage Table

Tx	Rx	Ch	Until
A	B	3	hh:mm:ss
C	D	1	hh:mm:ss

Any neighbor, upon receiving CFA or CFB, can calculate the expected duration of the data session as

$$Duration = TrainSize \times (T_{data} + SIFS) + T_{ack}, \quad (1)$$

where T_{data} and T_{ack} are the transmission time of a DATA (based on `PktSize`) and an ACK respectively. This duration will then be stored by the neighbor as an entry of its *channel usage table* shown in Table I.

IV. Hybrid Spectrum Sensing

As virtual spectrum sensing is complementary to the physical spectrum sensing approach, we design another protocol, VISH-II, based on hybrid spectrum sensing which combines both ideas. It adopts the same control session as VISH-I, but uses a redesigned packet train in the data session.

A. Fast Relinquishment with Reduced Interference

In general, there are two methods in acknowledging packet trains. One is *individual acknowledgment*: an ACK packet is sent for *each* successfully received DATA, as used by the IEEE 802.11 fragmented transmission [13]. The other is *batch acknowledgment*: only one ACK will be sent at the end of the sequence of DATA, as used by VISH-I and also like the TCP sliding window mechanism. Individual acknowledgment produces more interference due to the multiple ACK packets, and batch acknowledgment leads to wasted channel usage when two packet trains collide with each other or channel condition becomes bad.

VISH-II uses a packet train that overcomes the drawbacks of both methods; it releases a channel immediately when failure occurs, and reduces interference to nearby data sessions. See Fig. 5. Between every two consecutive DATA packets sent by a transmitter, an interval of TIFS (train inter-frame spacing) is inserted, during which the transmitter senses the data channel. If the channel is clear, it proceeds to send the next DATA. If the channel is busy (due to a nearby packet train transmission or the receiver sending NAK, explained shortly), it switches back

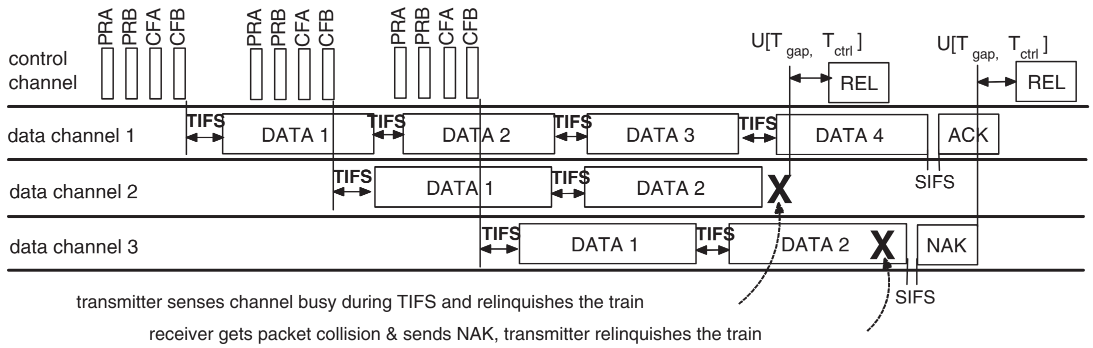


Figure 5: Data session of VISH-II. Physical sensing is performed during TIFS.

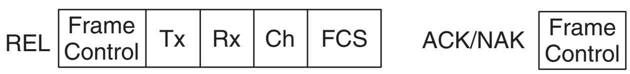


Figure 6: Additional frame format.

to the control channel immediately and thus *relinquishes* the train. For the receiver, it sends a NAK packet if it experiences DATA corruption (due to collision or noise), and then switches to the control channel. If it waits for DATA until timeout (due to the transmitter having relinquished the train), it switches to the control channel without sending NAK. The receiver keeps silent if a DATA is successfully received, and sends an ACK only if all DATA in the train are successfully received.

Accordingly, the calculation of a data session duration is changed from (1) to

$$\begin{aligned} Duration =& TrainSize \times (T_{data} + TIFS) \\ & - TIFS + SIFS + T_{ack}. \end{aligned} \tag{2}$$

TIFS is specified as

$$TIFS = SIFS + T_{plcp} + T_{turnard}, \tag{3}$$

where SIFS is for waiting for a possible NAK from the receiver, T_{plcp} is the transmission time of a PLCP (physical layer convergence protocol) preamble and header, used for CCA and packet header detection, and $T_{turnard}$ is the turnaround time of a transceiver, used for the transmitter to switch from RX to TX mode to send the next DATA.

After switching to the control channel, the transmitter or the receiver will send an REL (release) message to announce the advanced availability of the data channel. To avoid colliding with ongoing control sessions, it sends REL if the control channel is idle for $U[T_{gap}, T_{ctrl}]$, and cancels REL otherwise.

Compared to batch acknowledgment, this design of packet train can relinquish a train upon an intermediate failure and thereby avoid wasted channel occupancy. Compared to individual acknowledgment, this design allows for concurrent DATA receptions as no ACK will be sent between consecutive DATA packets. Also, because the interval between sending two consecutive DATA packets using individual acknowledgment is $2 \times SIFS + T_{ack}$ and

$$TIFS < 2 \times SIFS + T_{ack} \tag{4}$$

according to (3) ($SIFS > 2T_{turnard}$ and $T_{ack} > T_{plcp}$), this packet train takes shorter time to finish and hence reduces delay.

V. Performance Evaluation

In this section, we evaluate the performance of both protocols, i.e., VISH-I and VISH-II. In a sense, these protocols can also be viewed as multi-channel MAC protocols with priorities.

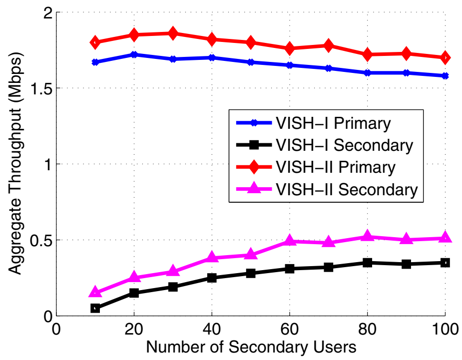
A. Simulation Setup

We use a Linux-based simulator developed using C++. Primary users and secondary users are randomly deployed in a network area of 1000m×1000m. The transmission range of each node is 200m and the interference range is 400m. The capture threshold is 6dB. There are 60 primary users. In each simulation, n nodes form $n/2$ disjoint flows (each node is either the source or the destination of a flow). Shortest path routing is used. Note that primary users route through primary users only and secondary users route through secondary users only.

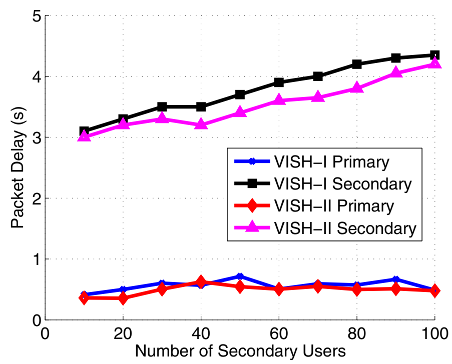
There are one control channel and five data channels with bandwidth 1Mbps each. The train size quota for primary and secondary users, Q^{pri} and Q^{sec}, are 5 and 4 respectively. Each source generates data packets with 1K-byte payload according to a Poisson point process. SIFS is $10\mu s$, $T_{turnard}$ is $3\mu s$, T_{ccap} is $35\mu s$, PLCP header is 6byte, PLCP (short) preamble is 9byte. The ID assignment is done in the initialization phase. Each simulation is terminated when a total of 50,000 data packets are sent over the network, and all results are averaged over 15 randomly generated networks.

B. Simulation Results

We assumes intra-group DISH for the two individual systems, i.e., primary users help primary users and secondary users help secondary users. On the other hand, inter-group DISH is asymmetric, i.e., secondary users help primary users but not vice versa.

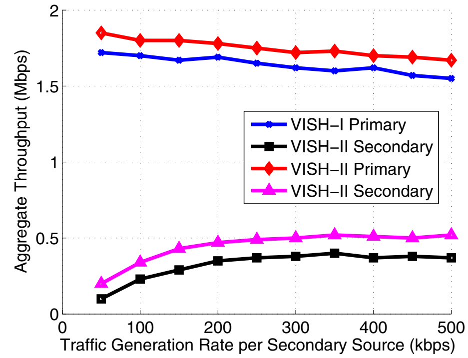


(a) End-to-end throughput.

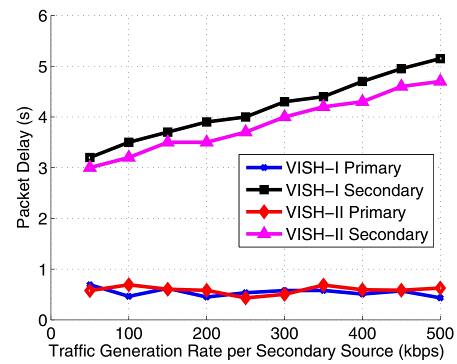


(b) End-to-end packet delay.

Figure 7: Quantifying "harmful interference" (varying N^{sec}). $\lambda^{pri} = 250kbps$ and $\lambda^{sec} = 100kbps$.



(a) End-to-end throughput.



(b) End-to-end packet delay.

Figure 8: Quantifying "harmful interference" (varying λ^{sec}). $N^{sec} = 40$ and $\lambda^{pri} = 250kbps$.

For the ease of description, we denote by N^{sec} the number of secondary users, and by λ^{pri} and λ^{sec} the traffic generation rate of a source primary and a source secondary user, respectively.

1) Result Set 1: Quantifying "Harmful Interference": To quantify the interference caused by secondary users to primary users, we measure end-to-end throughput and packet delay for primary users with respect to different numbers and traffic loads of secondary users. Two subsets of simulations are performed. In the first subset, we vary N^{sec} from 10 to 100, and show results in Fig. 7. We see in Fig. 7a that, although the number of secondary users increases and the throughput of secondary users grows by about 600% from $N^{sec} = 10$ to $N^{sec} = 100$, the throughput of primary users is maintained with only a slight downtrend (about 5% decrease from $N^{sec} = 10$ to $N^{sec} = 100$). In Fig. 7b, the packet delay of primary users is almost unchanged although the number of secondary users increases.

This conveys an important message: secondary users cause negligible interference to primary users despite that they are actually consuming the spectrum resource (their aggregate throughput increases). In addition, another observation is that VISH-II improves performance of VISH-I by 7%–50% at different N^{sec}, which justifies the idea of combining virtual sensing and physical sensing.

In the second subset of simulations as shown in Fig. 8, we vary λ^{sec} from 50kbps to 500kbps. The results are similar to the scenario of varying N^{sec} (Fig. 7) and confirm that, in both Cognitive DISH protocols, harmful interference caused by secondary users to primary users is well controlled.

2) Result Set 2: Overall Channel Utilization: As effective channel utilization translates to throughput, we measure the overall throughput summed over all primary and secondary users' flows with respect to different number of secondary users. In particular, we include the scenario of $N^{sec} = 0$, where there are primary users only.

In the results shown in Fig. 9, we see that the overall throughput at $N^{sec} = 0$ for VISH-I and VISH-II is merely 0.6 and 0.7 Mbps, respectively, but it sharply increases to 1.12 and 1.3 Mbps at $N^{sec} = 10$, indicating a remarkable 87% and 86% improvement, respectively. At $N^{sec} = 60$, this improvement reaches even higher to 203% and 200% for the two protocols. This observation demonstrates that both protocols well address channel under-utilization through opportunistically using spare spectrum, and significantly boost overall channel utilization.

3) Result Set 3: Effect of DISH: A special feature of the Cognitive DISH framework is that secondary users are not merely opportunistic *consumers* of system resources, but *contribute* to the system by providing DISH-based cooperation to primary users. To investigate the impact of this inter-

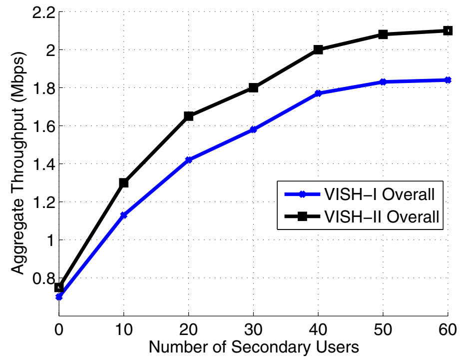
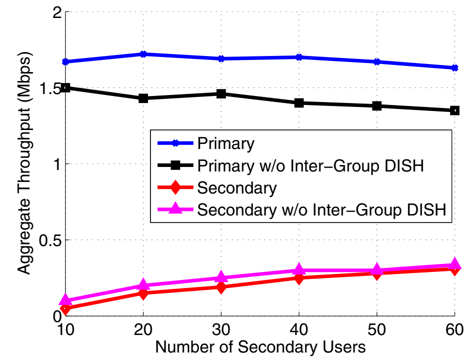
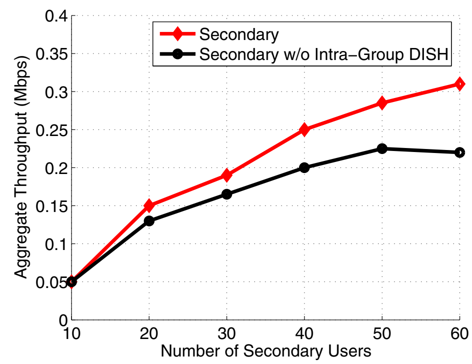


Figure 9: Overall channel utilization. $\lambda^{pri} = \lambda^{sec} = 100kbps$.



(a) The impact of inter-group DISH.



(b) The impact of intra-group DISH.

Figure 10: Investigating the impact of DISH. $\lambda^{pri} = 250kbps$ and $\lambda^{sec} = 100kbps$.

group DISH, we disable it in each protocol and then compare the performance. We also investigate intra-group DISH for secondary users by disabling cooperation among them. The results for VISH-I are given in Fig. 10. The results for VISH-II are similar and omitted.

Fig. 10a evaluates inter-group DISH. We see a clear gap (200–400kbps) between the throughput of primary users with and without inter-group DISH. On the other hand, the performance of secondary users is not notably affected. This demonstrates that the benefit from inter-group DISH is apparent and its associated overhead is negligible.

Intra-group DISH is evaluated in Fig. 10b. When there are 10 secondary users, there is no much difference between the two curves. This is because the channel contention is mild and hence channel conflict and deaf terminal problems rarely happen. However, as the number of users increases, the difference appears and a considerable 20–30% performance degradation is observed when without intra-group DISH.

In summary, we can conclude that both inter- and intra-group DISH are beneficial to system performance and hence is worthwhile to be considered in cognitive radio networks.

VI. Discussion

A. Security and User Reliability

Virtual spectrum sensing allows secondary users to know the activities happening in the primary user network. This seems to represent a user reliability and security concern because opening protocol details to external parties could be hazardous, as external (secondary) users may not be as reliable or benign as internal (primary) users.

In fact, this is not a problem specific to virtual spectrum sensing; it also applies to physical spectrum sensing. For instance, even if the protocol details are kept confidential to secondary users, a malicious user who does not understand the protocol can simply send jamming signals to each band to cause interference or channel congestion. Also, it is difficult to hide the use of any channel, regardless of virtual or physical spectrum sensing, because this can be easily detected via physically scanning all channels.

One feasible way to address security and privacy is to *encrypt* packets and make the encryption key only known to reliable users who have obtained permit. This would prevent malicious intruders from eavesdropping. This is out of the scope of this paper but is certainly worth future study.

B. Flexible Asymmetric DISH

VISH-I and VISH-II both use a simple “boolean” mode of asymmetric DISH, where secondary users cooperate with primary users but the converse is disabled. In fact, our framework described in Section II allows for flexible ways in implementing asymmetric DISH. In the below, we give a few extensions from the boolean scheme to a probabilistic scheme, where primary users can opt for helping secondary users according to the following:

1) a constant probability set at the discretion of individual primary users.
2) a probability p_{co}^{pri} computed on the fly based on the number of secondary neighbors, n^{sec}. These two quantities are inversely related (the larger n^{sec}, the lower p_{co}^{pri}), because a larger n^{sec} implies a higher likelihood for a secondary user to obtain *intra-group* DISH from other secondary users.
3) a more sophisticated probability computation such as Sift [17].

C. Multiple Cognitive Radios

In our current network model, each user has only one radio and hence will be blind to all other bands when it tunes its radio to any specific band. Suppose, on the other hand, that there are two radios available at each node. Then a user can monitor the spectrum usage when it tunes one of its radios to some band to perform data transmission or reception. In this case, the accuracy of user detection will be enhanced and false alarms would be significantly mitigated. This scenario is worth investigating.

VII. Conclusion

Conventional cognitive radio networks use physical spectrum sensing to detect the presence of primary users so that secondary users can opportunistically use the spectrum without causing harmful interference to primary users. In this paper, we propose virtual spectrum sensing as a complementary approach to perform user detection, which not only detects the presence but also the activities of primary users. This opens a broad range of possibilities, and we specifically incorporate DISH with virtual spectrum sensing to form a Cognitive DISH framework. Under this framework, we design two protocols using virtual spectrum sensing and hybrid spectrum sensing, respectively. Extensive simulations verify that (1) virtual spectrum sensing is a valid approach in that primary users are well protected against secondary users' interference and the overall channel utilization is significantly increased, (2) virtual sensing can be combined with physical sensing and the resultant hybrid sensing yields further performance improvement, and (3) DISH, either inter- or intra-group, has substantial benefit to cognitive radio networks and, particularly, inter-group DISH changes the role of secondary users from mere resource consumers to system contributors.

Acknowledgment

This work was supported in part by the National Research Foundation under grant NUS WBS No. R-705-000-018-279.

References

[1] FCC, "Spectrum policy task force report, ET Docket No. 02-155," Nov. 2002.

[2] J. Mitola III and J. Gerald Q. Maguire, "Cognitive radio: making software radios more personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, Aug 1999.

[3] Wireless@SG, http://www.singnet.com.sg/vas/.

[4] D. Cabric, S. M. Mishra, and R. W. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," in *Asilomar Conference on Signals, Systems, and Computers*, Nov. 2004, pp. 772–76.

[5] A. Ghasemi and E. S. Sousa, "Collaborative spectrum sensing for opportunistic access in fading environments," in *IEEE DySPAN*, Nov. 2005, pp. 131–36.

[6] ——, "Spectrum sensing in cognitive radio networks: Requirments, challenges and design trade-offs," *IEEE Communications Magzine*, pp. 32–39, April 2008.

[7] T. Luo, M. Motani, and V. Srinivasan, "CAM-MAC: A cooperative asynchronous multi-channel MAC protocol for ad hoc networks," in *IEEE Broadnets*, San Jose, CA, USA, October 2006.

[8] ——, "Cooperative asynchronous multichannel MAC: Design, analysis, and implementation," *IEEE Transactions on Mobile Computing*, vol. 8, no. 3, pp. 338–52, March 2009.

[9] ——, "Altruistic cooperation for energy-efficient multi-channel MAC protocols," in *ACM MobiCom*, Montreal, QC, Canada, 2007.

[10] ——, "Analyzing DISH for multi-channel MAC protocols in wireless networks," in *ACM MobiHoc*, Hong Kong, China, 2008.

[11] Q. Mahmoud, *Cognitive Networks: Towards Self-Aware Networks*. Wiley-Interscience, 2007.

[12] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: a survey," *Computer Networks*, vol. 50, no. 13, pp. 2127–2159, 2006.

[13] IEEE 802.11 Working Group, *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*, 1999.

[14] C.-C. Chen, E. Seo, H. Kim, and H. Luo, "Select: Self-learning collision avoidance for wireless networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 3, pp. 305–321, March 2008.

[15] D. Liu, Y. Zhang, and H. Zhang, "A self-learning call admission control scheme for CDMA cellular networks," *IEEE transactions on neural networks*, vol. 16, no. 5, pp. 1219–1228, 2005.

[16] D.-M. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," *Computer Networks and ISDN Systems*, vol. 17, no. 1, pp. 1–14, June 1989.

[17] K. Jamieson, H. Balakrishnan, and Y. C. Tay, "Sift: a MAC protocol for event-driven wireless sensor networks," in *Proceedings of the Third European Workshop on Wireless Sensor Networks (EWSN)*, February 2006, pp. 260–275.