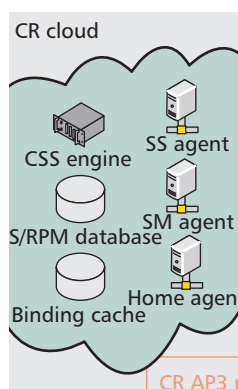


# A CLOUD MODEL AND CONCEPT PROTOTYPE FOR COGNITIVE RADIO NETWORKS

SAU-HSIUAN WU, HSI-LU CHAO, CHUN-HSIEN KO, SHANG-RU MO, CHUNG-TING JIANG,  
TZUNG-LIN LI, CHUNG-CHIEH CHENG, AND CHIAU-FENG LIANG,  
NATIONAL CHIAO TUNG UNIVERSITY



The authors propose a CR Cloud Networking model that is able to support CR access in TVWS. Making use of the flexible and vast computing capacity of the Cloud, a database and a cooperative spectrum sensing algorithm are realized on a CR Cloud (CRC) implemented with Microsoft's Windows Azure Cloud platform.

## ABSTRACT

The FCC's approval for the first commercial operation in TV white space gives new momentum to the development of cognitive radio in TVWS. On the other hand, the rapid growth of Cloud computing makes it possible and more economical to build a CR metropolitan area network with commodity hardware. In view of the opportunity and challenges brought about by these two technologies, we propose a CR cloud networking model that is able to support CR access in TVWS. Making use of the flexible and vast computing capacity of the cloud, a database and a cooperative spectrum sensing algorithm that estimates the radio power map of licensed users are realized on a CR cloud implemented with Microsoft's Windows Azure Cloud platform. The CRC can support CSS, dynamic spectrum access and mobility management. A medium access control protocol is also developed for this CRCN model to collect sensing reports and provide access to the TVWS and CRC services. Through this CRCN prototype, important network parameters such as the mean squared errors in CSS, the CR channel vacating delay, and the cloud-based handover time are measured for the design and deployment of the CRCN concept.

## COGNITIVE RADIO CLOUD NETWORKING SERVICE

The concept of cognitive radio (CR) was first introduced by Joseph Mitola III in [1]. Under the framework of CR, non-licensed users (secondary users, SUs) can access the channels for original licensed users (primary users, PUs) if the quality of service (QoS) to PUs can be effectively preserved or guaranteed. In view of the inefficient usages of some legacy spectrum holders (less than 25 percent overall according to [2]), CR is expected to resolve part of the spectrum shortage issues in wireless metropolitan and local area networks (MANs/LANs).

Over the past decade, extensive research has been done on CR in either theoretic explorations or circuit designs. The termination of

analog TV broadcasting allows the Federal Communications Commission (FCC) of the United States to take an initiative to test the CR concept in its TV spectrum in parts of its spaces when there are no TV's signal coverage (also referred to as TV white space, TVWS) [3]. The first TVWS service, provided by Spectrum Bridge Inc., is approved for field trials by the FCC in Wilmington, North Carolina, and authorized for commercial operations in January 2012. Encouraged by the FCC's CR policy, international organizations have started to define CR standards in TVWS (e.g., IEEE 802.22 and 802.11af).

Evolving from CR in TVWS, the concept has developed into a new paradigm of intelligent communications that encompass ideas such as spectrum sharing, competition, and reconfigurable and heterogeneous radio networks. These kinds of diverse and dynamic radio services rely on a computing infrastructure that is ubiquitous and elastic, yet has virtually unlimited resources to collect, analyze, process, and coordinate the massive and dynamic radio communications activities. Thanks to the rapid growth of cloud computing and fiber-to-the-home (FTTH) Internet services, one can start to imagine a future wireless city where seamless CR services provided by pervasive access points (APs) and cloud computing are as common as streetlighting services provided by ubiquitous lamps and power plants, as shown in Fig. 1. With such CR cloud networking (CRCN) service, the public can get seamless and free baseline wireless access to the *Internet avenue* just like walking in a city street at night without worrying about the light. As a matter of fact, many cities in the world (e.g. London, Beijing, and Taipei) have started to provide the public free Wi-Fi services at hot spots. To support dynamic and seamless CR access, as well as envisioning the great impact brought about by CRCN services, we present herein a primitive CRCN prototype and operating model that integrate functions of cooperative spectrum sensing (CSS), dynamic spectrum access (DSA), and mobility management on a unified cloud platform. The proposed CRCN infrastructure is illustrated in Fig. 2.

In another scenario where a regular Internet user wants to get connected to an SU in the CRCN, the user also needs to query the home agent of the CRC to establish the connection. In addition to these typical CR services, a Cloud-based mobility management service is also provided in the CRCN.

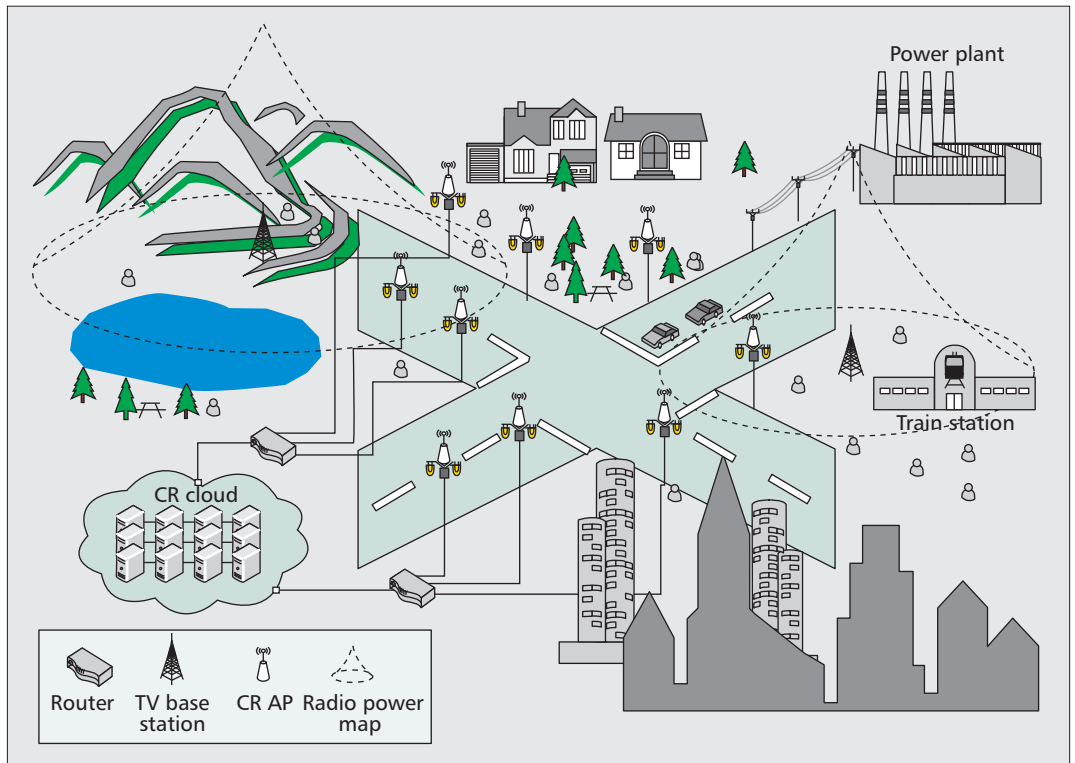


Figure 1. A cognitive radio cloud networking (CRCN) service model in TV white space.

### COGNITIVE RADIO CLOUD NETWORKING ARCHITECTURE

The CRCN architecture is mainly based on a CR cloud (CRC) and distributed CR APs that are connected to the CRC via an Internet backbone. The spectrum sensing (SS) agent at the CRC collects SS reports from CR APs or wireless sensing devices (SDs) that are associated with the APs, and drives a CSS engine to reconstruct the radio power map (RPM) of PUs' spectrum based on the SS reports and a sparse Bayesian learning (SBL) algorithm [4].

Upon requests for channel access from CR APs, the spectrum management (SM) agent at the CRC checks the RPM database and allocates appropriate spectra to the CR APs according to their locations and the spectrum white spaces in the RPM. The CR APs therefore coordinate both the SS reports from SDs as well as the DSA of SUs in the prototyped CRCN architecture. A CR medium access control (MAC) protocol that carries these two types of functions is introduced later in this article.

When connecting to the CRCN, an SU is first associated with a CR AP through a dedicated control channel. The CR AP then assigns, through the control channel as well, the SU a data channel and registers the SU to the home agent of the CRC. The home agent maintains a list of CR AP-SU pairs in the binding cache of the CRC. This binding cache allows the home agent to establish end-to-end connections between CR SUs. This type of channel access and data exchange process is illustrated with a packet delivery flow from SU2 to SU3 in Fig. 2. To ensure the access rights of PUs, the CR APs regularly check the SM agent about the avail-

ability of their data channels, and will reassign SUs' channels if necessary.

In another scenario where a regular Internet user wants to get connected to an SU in the CRCN, the user also needs to query the home agent of the CRC to establish the connection. In addition to these typical CR services, a cloud-based mobility management service is also provided in the CRCN. Specifically, when a CR AP receives packets that belong to an SU that is no longer associated with it, it will redirect the data packets to the home agent of the CRC. The home agent temporarily stores the data in the binding cache of the CRC. When the SU gets re-associated with a CR AP, the home agent updates its CR AP-SU list, and redirects the packets to the SU through its new associated AP. This cloud-based mobility management is illustrated with a data redirecting flow in the lower part of Fig. 2. The entire procedure for the cloud-based DSA and mobility management is specified in detail later.

### COMPARISONS OF COGNITIVE RADIO PROTOTYPES

To differentiate our proposed CRCN from existing CR prototypes, a Venn diagram in Fig. 3 is used to summarize and contrast the functions of various CR prototypes in the world. Since the proposal of the idea of CR [1], there have been numerous prototypes built to study feasible CR functions and services. What is shown in Fig. 3 is only an indicative and representative list due to space limitations. Four of them are prototypes by academia, and the other four are built by industry or government-funded research institutes. All these prototypes provide certain kinds of SS capabilities, and most consider ad-hoc CR networks, e.g. the *CRaMNET* by the University

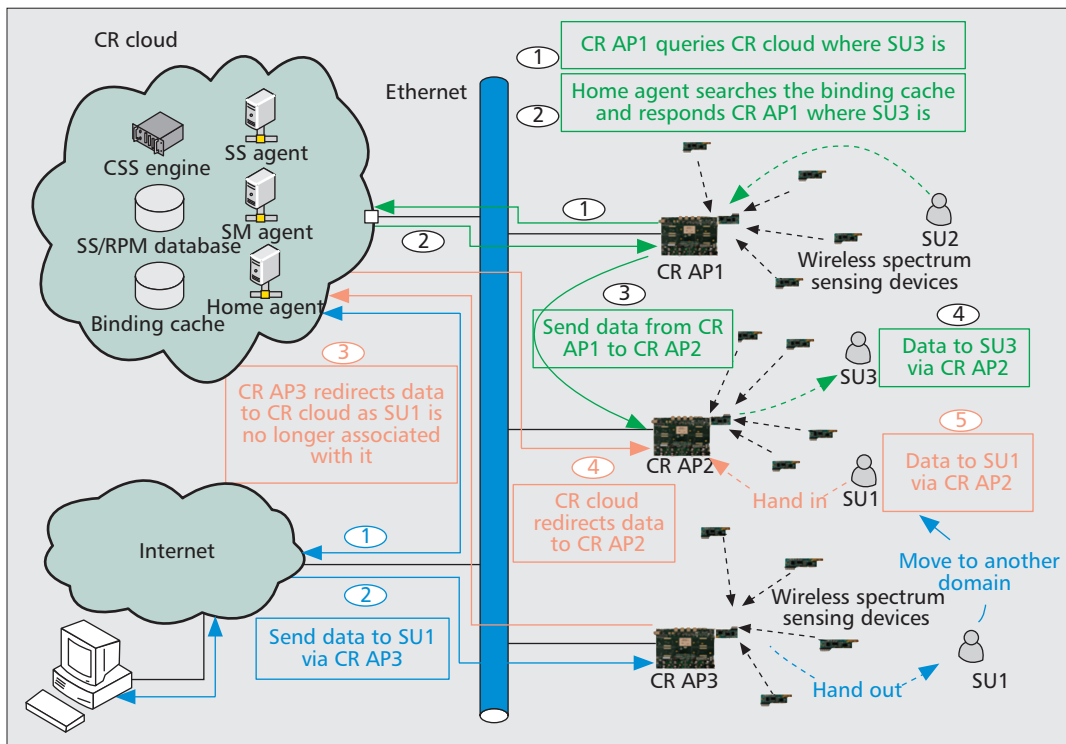


Figure 2. The prototyped CRCN architecture.

Taking CR in TVWS as an example, to ensure the received signal quality of TV sets, the FCC requires CR operators able to access databases that can provide the RPM of TV signals with a sensitivity of 0.8dBm below the noise level.

of Oulu, Finland [5], the *CogNet/Orbit* projects done by WINLAB, Rutgers University, United States [6], and other prototypes by Virginia Tech, United States [7] and Aachen University, Germany [8].

Even though there have also been some other prototypes or technologies for infrastructure-based CR networks, such as *Coral* designed by the Communications Research Centre, Canada [9], the Authorized Shared Access (ASA) service provided by Spectrum Bridge [10], and the cognitive wireless clouds (CWCs) proposed by Japan's National Institute of Information and Communications Technology (NICT) [11], a CR testbed that integrates the functions of CSS, DSA, and network and mobility management on a unified commercial cloud platform has not been developed to the best of our knowledge. These, however, are key functions for all the different types of CR prototypes in Fig. 3. In the CRCN service model illustrated in Fig. 1, one can imagine that a huge computing resource will be required to support the massive and dynamic interactions between CR SUs/SDs and the RPM database, CSS engine, and other network management units/agents of the CRC. The performance of a CRCN architecture like Fig. 2 will be determined by its scalability and the access time of cloud services, as well as the processing and networking delays inside the cloud. These motivate us to develop a CR prototype to study the feasibility of the CRCN concept. The prototype is developed based on Microsoft's Windows Azure to provide CRC services, and Rice University's Wireless Access Research Platform (WARP) [12] to implement the functions of a CR AP. An embedded Linux operating system is also ported onto the WARP to provide Internet

connectivity and realize a CR-MAC protocol. Important functions and features of our CRCN prototype are summarized below:

- A CRC implemented on Microsoft's Windows Azure
  - SS agent, database, and CSS engine to compute RPM and provide radio environment awareness
  - SM agent and RPM database for DSA of SUs
  - Home agent and binding cache for network and mobility management
- A CSS engine to reconstruct the RPM of PUs
  - A CSS algorithm based on a sparse Bayesian learning (SBL) algorithm
  - A MapReduce-like programming model to support scalable computation for CSS
  - Estimation of the number, locations, and radiated power propagation of PUs
- A cloud-based CR-MAC protocol
  - Coordination of SDs, and collection of SS reports
  - Frequency-division and time-division multiple access (FD/TDMA)
  - DSA of SUs
- A CR research platform that provides radio access, distributed SS and CRC services
  - CR APs, each integrating one control channel and seven data channels on a Linux-enriched WARP
  - A vector signal generator to generate digital TV signals in UHF band, 782~928 MHz
  - TI's CC1111 to provide control and data channels access, each with bandwidth up to 115.2 kHz
  - Hope radio frequency's (RF's) wireless SDs provide distributed readings of received signal strength indicators (RSSIs)

Considering that the positions and the population densities of CR SDs/SUs are supposed to be random and may vary in different times and areas, we develop a SBL-based CSS algorithm on Microsoft's Windows Azure Cloud platform.

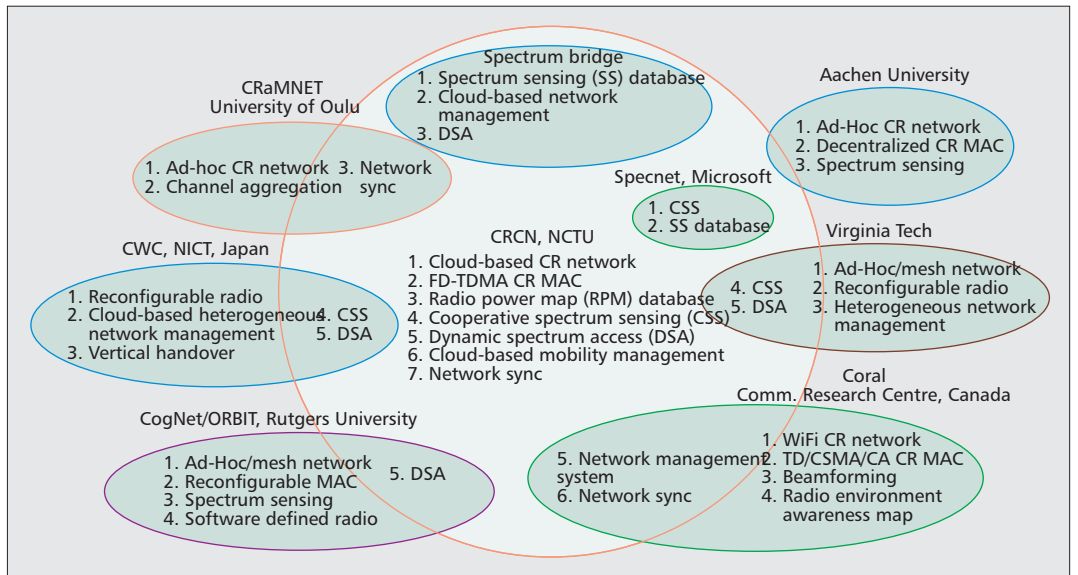


Figure 3. Function comparisons of various CR prototypes in the world.

In the sequel, we start to introduce the need for CSS and an SBL-based CSS algorithm that lays the foundation for CR services. A CR-MAC protocol that supports CSS and DSA in the CRCN is presented later, followed by a networking procedure designed to link the CR SUs/SDs, APs, and CRC services.

## A SPARSE BAYESIAN LEARNING METHOD FOR COOPERATIVE SPECTRUM SENSING

The CSS scheme implemented in this prototype is aimed to reconstruct the RPM of TV broadcasting or signals from rival CR operators. Taking CR in TVWS as an example, to ensure the received signal quality of TV sets, the FCC requires that CR operators be able to access databases that can provide the RPM of TV signals with a sensitivity of 0.8 dBm below the noise level (-106.2 dBm). In addition, a database also needs to provide the geographical locations of TV base stations (BSs), their heights above the terrain, antenna patterns, radiation powers, channel numbers, and call signs. SUs use the databases to determine their access rights in TVWS. To achieve these multiple purposes of CR databases in TVWS, some research units (e.g., Microsoft Labs) have started to develop databases (Specnet) that make use of location-based CSS.

Although the geographical locations and some other parameters of TV BSs are stationary and may be available, it remains difficult to reconstruct the composite RPM of individual BSs that are broadcasting signals, as well as their corresponding radiation powers and propagation decays in different places. The problem becomes even more challenging when it comes to estimate the PRMs of more dynamic rival CR operators. Therefore, a CSS scheme that makes use of SS reports collected from distributed SDs or SUs are in fact necessary. The reasons are at least twofold: First, the radiation power of a TV BS typically covers a much larger area which is beyond the reach of the transmit power of a

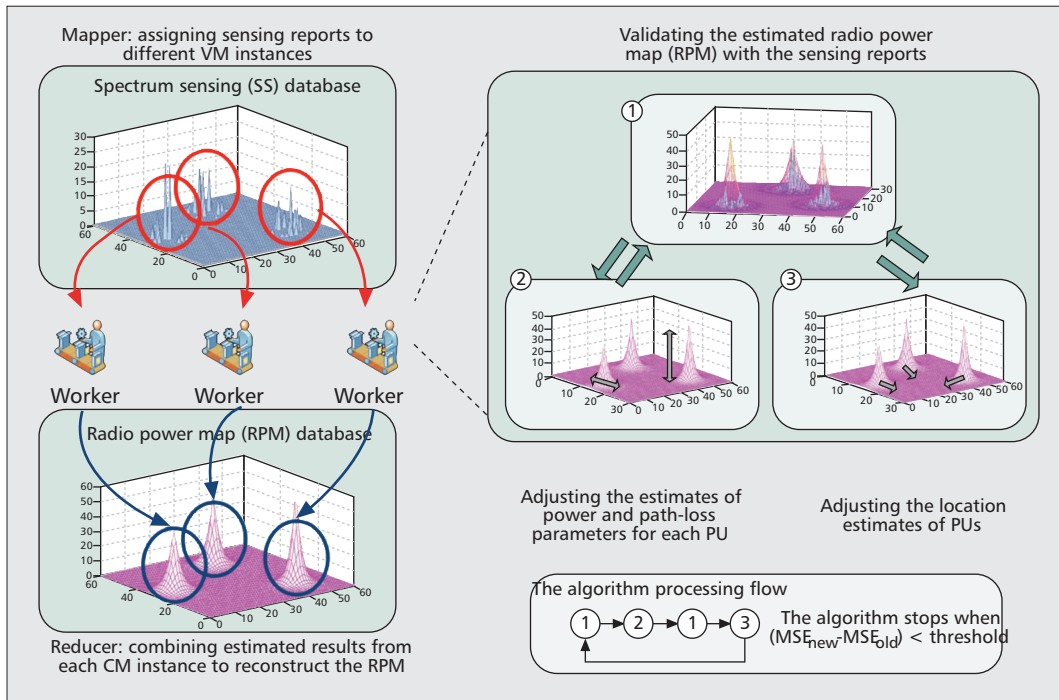
sensing or mobile device. To reconstruct the RPM of even a TV BS requires sensing reports from SDs located in different places inside the TV signal's coverage range. Considering the much weaker transmit power (50 mW) of a TVWS SD, it requires many CR APs in the same area to collect the sensing reports from distributed SDs. The CR APs then send the SS reports to the SS database on the CRC for CSS engines to reconstruct the RPM of the TV signal.

Second, it is also necessary to estimate the signal coverage ranges of CR APs and SUs from rival CR networks to make the most out of TVWS. Even if the signal coverage ranges of such TV band devices (TVBDs) are much smaller than that of a TV BS, they are more dynamic and overlapping to some extent such that their composite RPMs may have different shapes and vary frequently. These two important features of CR in TVWS have attracted significant research attention in CSS. To handle the massive SS reports and overcome the algorithm complexity in reconstructing the RPMs of TV BSs or TVBDs, we propose a computational framework for CSS under the CRCN architecture in Fig. 2. The CSS method only requires the positions and RSSI readings collected from SDs or SUs. Considering that the positions and population densities of CR SDs/SUs are supposed to be random and may vary in different times and areas, we develop an SBL-based CSS algorithm on Microsoft's Windows Azure cloud platform. With the scalability of cloud computing, the proposed method is in principle able to do CSS with sparse sensing reports over a huge area. More details about the operating procedure of the CSS algorithm are provided below [4, 15].

## CSS WITH SPARSE AND RANDOMLY DISTRIBUTED SS REPORTS

Assume that there are  $N$  SDs or SUs in an area of  $N_p \times N_p$  where there are  $M_p$  active PUs. The RSSI vector  $\mathbf{r} = [r_1, r_2, \dots, r_N]^T$  collected from the SDs or SUs at the positions of  $\mathbf{Z} = [z_1, z_2, \dots, z_N]$





**Figure 4.** Illustrations of a sparse Bayesian learning procedure for RPM reconstruction.

in this area, with  $z_i = [x_i, y_i]^T$  being the 2D coordinates of the  $i$ th RSSI reading, are fed back to the CRC by their associated CR APs. The RSSI vector  $\mathbf{r}$  in decibels (dB) is modeled in a linear regression form of

$$\mathbf{r} \triangleq \Psi \mathbf{w} + \mathbf{u}, \quad (1)$$

where  $\mathbf{w}$  is an  $M \times 1$  weighting vector, and  $\Psi = [\boldsymbol{\varphi}_1(\mathbf{Z}), \dots, \boldsymbol{\varphi}_M(\mathbf{Z})]$  is a  $N \times M$  matrix defined by the values  $\boldsymbol{\varphi}_j(\mathbf{Z}) \equiv [\boldsymbol{\varphi}_j(z_1), \dots, \boldsymbol{\varphi}_j(z_N)]^T$  of the basis functions  $\boldsymbol{\varphi}_j, j = 1, \dots, M$ , evaluated at the locations of  $z_1, z_2, \dots, z_N$  from which the  $N$  RSSI readings are collected.

This regression model 1 is intended to fit the  $N$  collected RSSI readings with a weighted linear combination of the  $M$  basis vectors,  $\boldsymbol{\varphi}_j(\mathbf{Z})$ ; thereby one can estimate the RPM,  $\Psi \mathbf{w}$ , and hence the number of active PUs, their corresponding positions, radiation powers, and propagation loss models. More specifically, the basis functions,  $\boldsymbol{\varphi}_j(z_i), j = 1, \dots, M$ , are defined in a form of

$$\boldsymbol{\varphi}_j(z_i) \triangleq \frac{1}{2s_j} \exp \left\{ -\frac{\sqrt{(x_i - c_{j,x})^2 + (y_i - c_{j,y})^2}}{s_j} \right\} \quad (2)$$

attempting to approximate the standard radiation power loss model of

$$P_r \triangleq P_t - \overline{PL}(d_0) - 10n[\log(d) - \log(d_0)] \text{ (dB)} \quad (3)$$

with adjustable parameters  $\mathbf{c}_j \equiv [c_{j,x}, c_{j,y}]^T$  and  $s_j$  that characterize the locations and power decaying rate of PUs' signal sources. The parameter  $P_r$  in Eq. 3 is the signal power received in dB at a propagation distance,  $d$ , from the signal source;  $P_t$  stands for the source radiation power;  $\overline{PL}(d_0)$

is the signal power observed at a reference distance,  $d_0$ , and  $n$  is the path loss exponent at which the radiation power decreases with the propagation distance  $d$ . On the other hand, the  $N \times 1$  uncertainty vector  $\mathbf{u}$  models the logarithms of the shadowing effects at the  $N$  different collection sites, and is thus considered a zero-mean and white Gaussian random vector with variance equal to  $\beta^{-1}$ , denoted by  $\mathcal{N}(0, \beta^{-1} \mathbf{I}_N)$ .

The objective of CSS is therefore to estimate the weighting vector,  $\mathbf{w}$ , the parameters  $\mathbf{C} = [c_1, \dots, c_M]$  and  $\mathbf{s} = [s_1, \dots, s_M]$  of the basis functions, and the exact values of  $M$  from the sparse measurements  $\mathbf{r}$  collected at the locations  $\mathbf{Z}$ . If the estimates are accurate enough, one can expect that approximates the number of active PUs that transmit signals at  $\hat{c}_j, j = 1, \dots, M$ , with radiation power equal to  $\hat{w}_j/2s_j$  (dB), respectively.

The entire problem appears to be a complicated estimation problem, and is thus partitioned into several subproblems to approach the solutions in an alternative and iterative manner under the Bayesian inference framework [4]. Specifically, the estimation problem is executed in three steps:

1. Estimate  $\mathbf{w}$  and  $\beta$ , given the estimates of  $\hat{\mathbf{C}}, \hat{\mathbf{s}}$ , and  $\hat{M}$ .
2. Delete unnecessary bases and update the estimate  $\hat{M}$  based on  $\hat{\mathbf{C}}, \hat{\mathbf{s}}$ , and the new estimates  $\hat{\mathbf{w}}$  and  $\hat{\beta}$ .
3. Adjust  $\hat{c}_j$  and  $\hat{s}_j$  of the survival bases, given  $\hat{M}, \hat{\mathbf{w}}$  and  $\hat{\beta}$ , and stop if  $\ln\{P(\mathbf{r} | \hat{\mathbf{w}}, \hat{\mathbf{C}}, \hat{\mathbf{s}}, \hat{\beta}, \hat{M})\}$  converges. Otherwise, go back to Eq. 1.

Under the CRCN architecture in Fig. 2, the SBL-based CSS algorithm is implemented on Windows Azure with a MapReduce-like programming model [14, 15]. The input SS data from CR APs are first processed by the SS agent and stored in different tables of the SS database according to their locations. The CSS

The cloud-based CR MAC protocol in this prototype aims at collecting SS reports and coordinating channel accesses among SUs and SDs. From the aspect of commercialization of TVWS and CR, one big step, but not the only step, is the progress of the IEEE 802.11af standard.

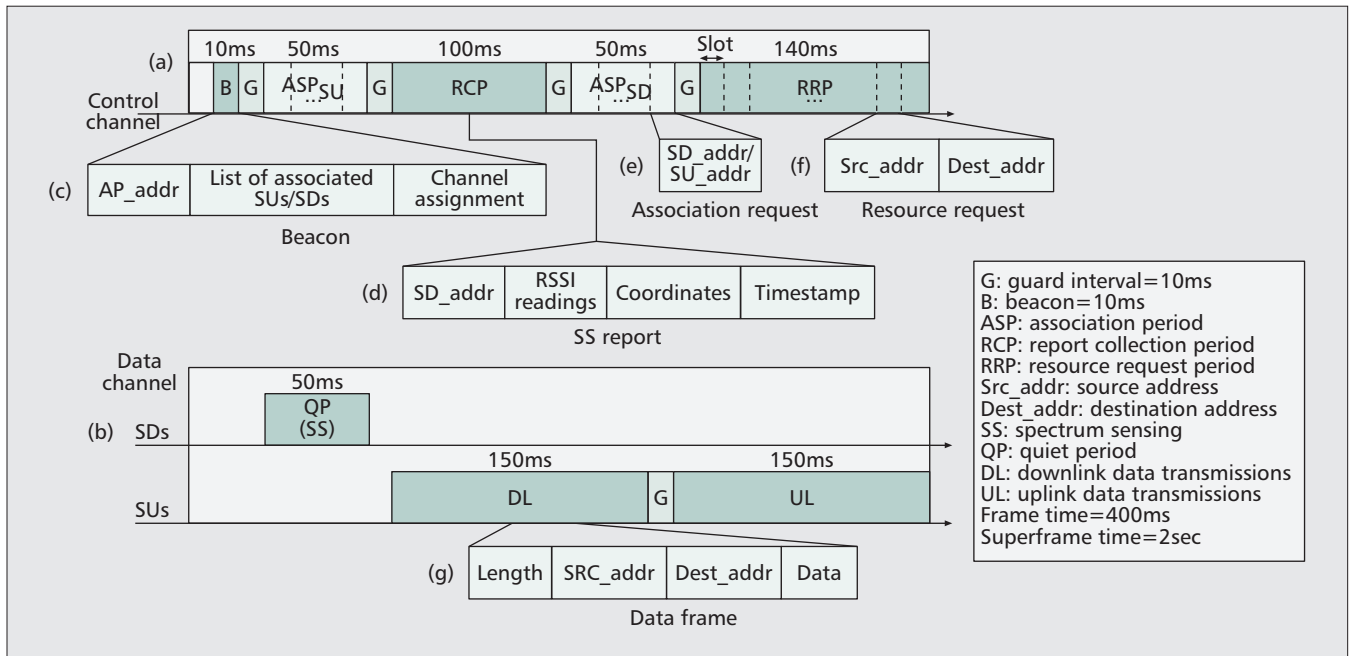


Figure 5. Framing and the five defined frame formats of the proposed CR MAC protocol.

engine, which is formed with a set of virtual machines (VMs), is activated periodically by the SS agent to estimate  $\hat{w}$ ,  $\hat{C}$ ,  $\hat{s}$ ,  $\hat{\beta}$ , and  $\hat{M}$  in parallel with data in each table, following steps 1 to 3. The aforementioned estimates are stored back to the RPM database, and then used to reconstruct the RPM,  $\Psi(\hat{C}, \hat{s})\hat{w}$ , to be used by the SM agent. The entire procedure is illustrated in Fig. 4.

## CLOUD-BASED CR MAC PROTOCOL

The cloud-based CR MAC protocol in this prototype aims at collecting SS reports and coordinating channel accesses among SUs and SDs. From the aspect of commercialization of TVWS and CR, one big step, not the only step, is the progress of IEEE 802.11af standard. Taking IEEE 802.11af as an example, on the TVWS side, channel bandwidth is 6 MHz, and available channels may not be contiguous; on the WiFi side, associated users share a channel in a time-division multiple access manner. We, thus, design and implement a frequency-division and time-division multiple access (FD-TDMA) MAC protocol for our CRCN prototype.

In our design, TVWS channels assigned to a CR AP are partitioned into one control and multiple data channels. The control channel is for the CR AP to collect sensing reports and coordinate data channel accesses; the data channels are assigned to SUs for data deliveries. Multiple SUs can be assigned to different data channels or share the same channel in a TDMA manner. SDs and SUs always listen to the control channel unless they are requested to perform channel sensing or granted to deliver data. In the following, we first introduce the frame structure designed for CSS and channel access, followed by the detailed operations.

## FRAMING AND FRAME FORMATS

In the proposed CR-MAC, time is framed, and several frames are grouped into a superframe. For simplicity, we only show one data channel in a time frame in Fig. 5. Each frame on the control channel consists of five fields: Beacon (B), Association Period for SUs ( $ASP_{SU}$ ), Report Collection Period (RCP), Association Period for SDs ( $ASP_{SD}$ ), and Resource Request Period (RRP), as shown in Fig. 5a. In our implementation, considering the hardware constraints and synchronization precision, we insert a 10ms guard time between two consecutive fields. On the other hand, a frame on the data channel has three fields: Quiet Period (QP), DownLink data transmissions (DL), and UpLink data transmissions (UL), as shown in Fig. 5b. Similarly, a guard time is inserted between DL and UL fields.

In our design, CR APs periodically broadcast beacons on the control channel to announce their existence and manage channel activities, and SUs or SDs must associate and synchronize with a CR AP before accessing channels or performing SS. As a result, a control frame starts with a beacon, which consists of three fields: CR AP address ( $Addr_{AP}$ ), the list of associated SUs and SDs, and data channel assignments (Fig. 5c). Moreover,  $ASP_{SU}$  and  $ASP_{SD}$  are for SUs and SDs to send association requests to their intended CR APs, respectively. Both fields are divided into time slots, and adopt a contention-based multiple access scheme.

Each time slot of  $ASP_{SU}$  or  $ASP_{SD}$  exactly accommodates an association request, which contains the issued SU's or SD's address (Fig. 5e). The CR AP utilizes the RCP to poll the associated SDs for SS results, while SDs utilize the RRP to send resource requests to the CR AP. A sensing report contains the information of the SD's address, measured RSSI readings of

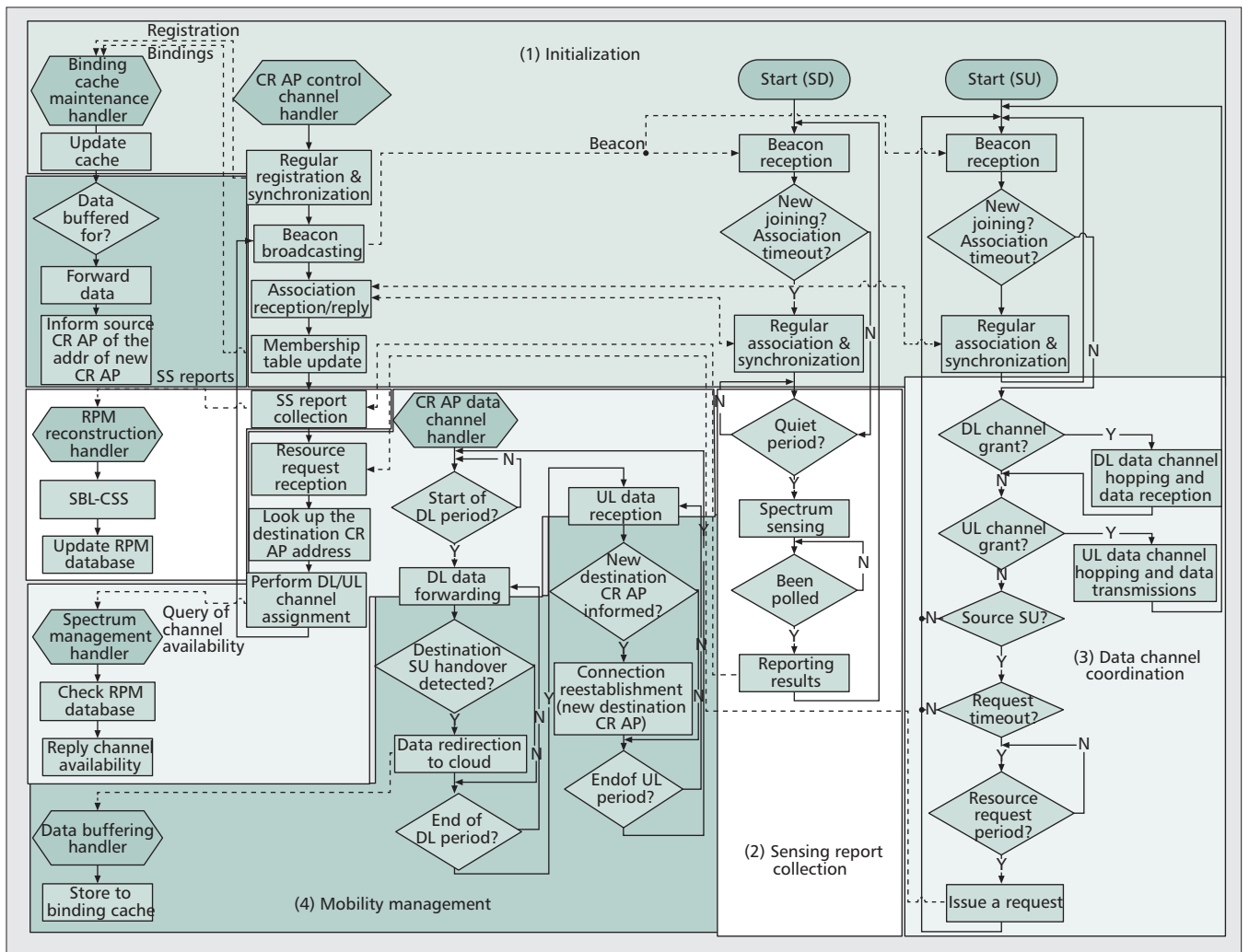


Figure 6. The flowchart of the designed cloud-based CR-MAC protocol.

data channels, coordinates, and timestamp (Fig. 5d). Similarly to  $ASP_{SU}$  and  $ASP_{SD}$ , the RRP is time-slotted, and a resource request contains the issuing SU's address, denoted  $Src\_addr$ , and the intended SU's address, denoted  $Dest\_addr$ , as shown in Fig. 5f.

Upon association, an SD hops to and senses data channels sequentially in the QP. To accurately detect PUs' activities on data channels, all SUs are forbidden to transmit data in this time duration. Following the QP, the DL and UL periods are, respectively, for a CR AP to transceive data to and from it scheduled SUs. A data packet consists of packet length in byte (Length),  $Src\_addr$ ,  $Dest\_addr$ , and data payload (Data), shown in Fig. 5g. The defined acronyms and abbreviations in the CR-MAC are summarized in a table in Fig. 5 as well.

### PROTOCOL OPERATIONS

The proposed cloud-based CR-MAC protocol coordinates network components of CRC, CR APs, SDs, and SUs to complete the four functioning phases of the CRCN, including initialization, SS report collection, channel access coordination, and cloud-based mobility management. To describe and depict the intertwined relationship between the network components

and functions, the control flows of different components are shown in separate flowcharts in Fig. 6, each started with a handler. Furthermore, the functioning phases, which involve interactions among multiple components, are marked with different background colors in the figure, and separated from each other with small spacings. The solid lines represent the control flow in a management function of a network component, while the dashed lines depict the relationship of triggering events between network components. The details of the four functioning phases are described below.

**Initialization** — The initialization procedure is summarized in Fig. 6 (1). When connecting to the CRCN, CR APs, SUs, and SDs first perform registration/association and time synchronization. Indeed, CR APs send registration requests to the home agent in the CRC. Upon receiving registration requests, the home agent updates the binding cache correspondingly, and replies to the CR APs with the timestamp. The registered CR APs now are able to broadcast beacons and start to serve SUs.

For a new joining SU (or SD), when hearing a beacon, it will send an association request to the CR AP in a randomly selected time slot of

To establish the end-to-end data path, the source CR AP then looks up its membership table to discover whether the intended destination SU is also associated with itself. The source CR AP also performs DL channel assignment if the destination SU is associated with itself.

the  $ASP_{SU}$  (or  $ASP_{SD}$ ). Similar to the CRC, the CR AP maintains a membership table to record the associated SUs' and SDs' addresses. It further forwards the address information of its associated SUs to the home agent for communication establishment and mobility management. Note that the association relationship between CR APs and SUs is also maintained in the binding cache. The association process is successful if the SU (or SD) is listed in the "associated SUs and SDs" of the next beacon. Otherwise, the SU or SD keeps sending association requests in the following control frames. After being associated, SDs and SUs can start to perform SS and request resources on demand, respectively, under the coordination of the CR AP.

Both membership table and binding cache adopt a "soft-state" approach for dynamic membership and network topology changes. That is, CR APs, SUs, and SDs periodically send registration/association requests to refresh their membership. The corresponding entries in the membership table and binding cache will be removed after timeout.

**Sensing Report Collection** — The procedure of SS report collection is summarized in Fig. 6 (2). Associated SDs synchronously and sequentially sense all data channels in the QP. Note that in the QP, although associated SUs cannot transmit data on data channels, new joining SUs are allowed to perform association on the control channel. After sensing, the SDs hop back to the control channel and wait for the CR AP's polling to report their sensing results in the RCP. The reason for adopting proactive polling is that SDs are deployed for SS and reporting only; thus, a polling mechanism can significantly increase the flexibility of SD sampling and the accuracy of RPM reconstruction.

The CR AP forwards the gathered sensing reports to the SS agent in the CRC via the Internet backbone. The SS agent then drives the CSS engine to derive the RPM of PUs and updates the RPM database.

**Channel Access Coordination** — Although associated, an SU cannot access any data channel unless it gets a grant. Therefore, when an associated SU wants to initiate a data communication (in the following, we call it source SU), it first sends a resource request, which is with a timer, to its CR AP (here we name it source CR AP) in a randomly selected time slot of the RRP. The source CR AP then queries the SM agent in the CRC about the available data channels. The SM agent, after checking the RPM database, replies to the source with the CR AP the allocated data channels. The source CR AP then performs channel assignment for UL transmissions.

To establish the end-to-end data path, the source CR AP then looks up its membership table to discover whether the intended destination SU is also associated with itself. The source CR AP also performs DL channel assignment if the destination SU is associated with itself. It informs both the source and destination SUs of the allocated channels in the next beacon.

On the other hand, if the destination SU is not under its coordination, the source CR AP

queries the home agent in the CRC about the destination SU's associated CR AP (here we name it destination CR AP). Upon knowing the address of the destination CR AP, the source CR AP not only notifies the source SU of the assigned data channel for UL transmissions, but also forwards received data frames to the destination CR AP. The destination CR AP is responsible to further deliver data frames to the destination SU by allocating a DL data channel. It may occur that the intended destination SU is unknown to the CRCN. In such a situation, the source CR AP will not grant the source SU's resource request. The procedure of sensing report collection is summarized in Fig. 6 (3).

To protect the access rights of PUs, CR APs regularly query the SM agent about the availability of their allocated data channels. SUs vacate the assigned data channels when either finishing data transmissions or PUs are detected. For the latter, the CR APs will assign another available data channels to the SUs for the remaining data transmissions.

**Cloud-Based Mobility Management** — Considering that a destination SU re-associates with another CR AP (in the following, we call it new destination CR AP) due to mobility, the source CR AP is not aware of this change, and keeps forwarding data frames to the original destination CR AP (here we name it "old destination CR AP"). One unique feature of the prototyped CR-MAC protocol is the cloud-based mobility management.

The procedure of mobility management is summarized in Fig. 6 (4). Because of the "soft-state" table maintenance approach, the old destination CR AP will discover that the destination SU is no longer under its management. It then redirects all received data frames, which are destined to the destination SU, to the home agent in the CRC for temporarily buffering. When the link of the new destination CR AP and the destination SU is updated, the home agent delivers all buffered data frames to the destination SU. Besides, the home agent informs the source CR AP of the address of the new destination CR AP. From now on, the source CR AP delivers data frames directly to the new destination CR AP.

## A CRCN PROTOTYPE AND EXPERIMENT RESULTS

Experimental results on the CRCN prototype are presented in this section. The experimental setting is as shown in Fig. 2. The CRC is implemented on Windows Azure, with the databases built with SQL Azure. On the other hand, CR APs are implemented with WARPs [12], and SUs and SDs use TI's transceiver modules CC1111 and the HM-TR transceiver modules of HOPERF [13], respectively, to communicate with WARPs. There are 10 SDs, 2 SUs, and 2 CR APs in the experimental CRCN prototype.

To avoid interfering with TV signals, the frequency channels used in experiments are set in the industrial, scientific, and medical (ISM) bands right above the UHF TV band. The con-



control channels used for each CR AP are set at the 866 and 868 MHz bands, and the data channels are set at the 870 and 872 MHz bands with data rates equal to 115.2 kb/s.

In this CRCN prototype, the frame time is 400ms; a superframe consists of five frames so that its duration is 2 s. The time settings for each field in a control frame and a data frame are shown in Fig. 5. The HM-TR of an SD spends 20 ms on sensing a data channel; thus, it can sense two data channels in a QP. Besides, a CR AP can poll five HM-TRs to collect reports and forward the collected reports to the SS agent in a frame. In our design, a CR AP queries the SM agent about the availability of its allocated data channels once per superframe (indeed, at the beginning of the last frame in a superframe). The superframe time is set to meet the FCC's requirement to vacate a channel within 2 s when its PU is detected.

### ROBUSTNESS OF THE PROPOSED SBL-BASED CSS SCHEME

In the experiments, there is only one PU BS in a designated area. The PU's frequency is set at 870 MHz. And SSs are done by SDs whose RSSI readings only consist of five strength levels, corresponding to:

- Level 1: -96.2 dBm~-87.1 dBm
- Level 2: -87.1 dBm~-78.6 dBm
- Level 3: -78.6 dBm~-71.2 dBm
- Level 4: -71.2 dBm~-61.0 dBm
- Level 5: -61.0 dBm~

To conduct experiments in a constrained lab space, the transmission power of the PU BS is set to the same order of the CR AP such that the RPM of a PU BS can be reconstructed with the sensing reports of the 10 SDs. This setting not only simplifies the complexity of experiments, but also helps assess the robustness of the proposed SBL-based CSS algorithm as there are now only 10 SDs whose RSSI readings are partitioned only into 5 levels in a wide range of signal strengths from -96 to -61 dBm. Experimental results show that the mean squared error in the estimated PU location becomes negligible when the number of SDs exceeds 8. And the missing ratio is less than 0.1 when the number of SDs is greater than 4. Both results show that the proposed SBL-based CSS algorithm is quite robust even with a limited number of imprecise sensing measurements.

### RESPONSE TIME OF CHANNEL VACATING IN CRCN

Channel vacating time is the elapsed time from the presence of a PU to the SU's release of the PU's occupied channel. Several factors may affect the channel vacating time, such as network communication delays, the time to read/write databases, and the execution time of the CSS algorithm. Based on our measurements, the minimum, average, and maximum round-trip network delays between the CR AP and the CRC are (4.1, 4.2, 5.7) ms, respectively. Besides, the three-tuples of the (minimum, average, maximum) computation times for performing the CSS algorithm are (0.27, 0.54, 0.91) s, respective-

ly. The three-tuples of spending times for the SS agent to write the SS database and for the SM agent to read the RPM database are (0.15, 0.31, 0.66) s and (0.11, 0.31, 0.92) s, respectively. As a result, the CRC processing delays, which are sums of these above network and processing delays, fall within [0.5341, 2.4957] s.

In addition to the CRC processing delays, channel vacating time also depends on PUs' presence times as SS is performed once per superframe. More specifically, the SS reports are sent back by the CR AP in the end of the RCP of each frame (Fig. 5a). Besides, the CR AP queries the CRC for channels' availability at the end of the beacon period of the last frame of a superframe, and updates its channel usage in the first beacon period at the beginning of a superframe. Consequently, if a PU appears right before the first QP of a superframe, and its radio signal gets sensed and processed by the CRC right after the first RCP with a minimum amount of time (0.5341 s), the CRC is aware of the PU's presence before the CR AP's inquiry in the very same superframe. As a result, the PU's channel will be released by the CR AP in the first beacon period of the next superframe, ending up with a shortest channel vacating time of 1.98 s.

On the contrary, if a PU appears right after the first QP of a superframe, its radio signal will be sent and processed by the CRC only after the first RCP of the next superframe. The waiting time will become 2.11 s. Moreover, if the CRC in this case happens to spend a maximum amount of time (2.4957 s) in processing the SS reports, the CRC will not be aware of the PU's presence before the CR AP's inquiry for channel status in the same superframe. As a result, the PU's channel will not be released by the CR AP until the first beacon of the fourth superframe, ending up with a longest channel vacating time of 5.92 s. We note that in practice, the overall channel vacating time might also be slightly affected by the processing time of the CR AP.

### THE CLOUD-BASED HANDOVER TIME IN CRCN

In this experiment, we estimate the time spent to hand over data destined to a mobile SU from one CR AP to another. The handover time is mainly determined by the round-trip network delays between APs and the CRC, whose three-tuples of (min, average, max) times are (4.1, 4.2, 5.7) ms; the time to save the redirected data to the binding cache and update the offline status of the intended SU at the home agent, whose (min, average, max) times are (62.5, 70.8, 93.7) ms; as well as the time to retrieve the redirected data from the binding cache and look up the address of the SU's new associated CR AP, whose (min, average, max) times are (151, 204, 343.7) ms. As a result, ignoring the processing time for CR APs to handle the data, the minimum, average, and maximum handover times out of 1000 runs are 218 ms, 279 ms, and 443 ms, respectively.

Based on the experimental results, we observed that in general cases, the measured handover time matches the 400 ms frame time. That is, if an SU moves away from one CR AP and gets re-associated with another CR AP in a

Channel vacating time is the elapsed time from the presence of a PU to the SU's release of the PU's occupied channel. Several factors may affect the channel vacating time, such as the network communication delays, the time to read/write databases, and the execution time of the CSS algorithm.

Despite the fact that the true CRCN system performance will only be available after large volume tests, the experimental results did reveal that the proposed CRCN concept is a promising technique that might open up a new CR networking model.

MAC frame, the CRC can redirect data from its previous CR AP to its current one in the next MAC frame. Moreover, we found that data retrieval and address lookup times dominate the worst case handover time. To reduce the worst case handover time in order to match the frame time, we may employ a more efficient searching algorithm in the CRC.

## CONCLUSIONS

We introduce a concept prototype of a CRCN model and evaluate the time to perform CSS, CR channel vacating, and handover in a CRC implemented with Microsoft's Windows Azure cloud platform. Experimental results showed that the proposed SBL-based CSS algorithm performs well even with a limited number of imprecise sensing measurements. Although the overall system response time is slightly longer than the FCC's policy to access the TVWS, it could effectively be improved with some system adjustments and more advanced searching algorithms. Despite the fact that the true CRCN system performance will only be available after large volume tests, the experimental results did reveal that the proposed CRCN concept is a promising technique that might open up a new CR networking model. More advanced studies such as scalable cloud-based CSS schemes and CR resource management methods remain crucial to justify the feasibility of such a CRCN concept in the support of large-volume public access.

## ACKNOWLEDGEMENT

This work was supported in part by the National Science Council, Taiwan under Grant NSC 99-3113-P-009-004.

## REFERENCES

- [1] J. Mitola and G. Q. Maguire, "Cognitive Radio: Making Software Radios More Personal," *IEEE Pers. Commun.*, vol. 6, no. 4, Aug. 1999, pp. 13–18.
- [2] M. A. McHenry, *NSF Spectrum Occupancy Measurements Project Summary*, Shared Spectrum Co. Report, Aug. 2005.
- [3] FCC, Second Report and Order and Memorandum Opinion and Order, FCC 08-260, Nov. 2008.
- [4] D.-H. T. Huang, S.-H. Wu, and P.-H. Wang, "Cooperative Spectrum Sensing and Locationing: A Sparse Bayesian Learning Approach," *Proc. IEEE GLOBECOM*, Miami, FL, Dec. 2010.
- [5] R. Sathyanarayana et al., "CraMNET: Opportunistic Cognitive Radio for MANET with Adaptive PHY and Dynamic Routing Capability," *SDR-09 Technical Conf.*, Washington DC, Dec. 2009.
- [6] D. Raychaudhuri et al., "CogNet — An Architectural Foundation for Experimental Cognitive Radio Networks Within the Future Internet," *Proc. ACM MobiArch*, Dec. 2006.
- [7] F. Ge et al., "A Heterogeneous Cognitive Radio Network Enabling Dissimilar Cooperative Spectrum Sensing, Dynamic Spectrum Access, and Interoperability," *Proc. IEEE DySPAN*, Chicago, IL, Oct. 2008.
- [8] J. Ansari et al., "A Flexible MAC Development Framework for Cognitive Radio Systems," *Proc. IEEE WCNC*, Quintana Roo, Mexico, Mar. 2011.
- [9] J. Sydor et al., "Cognitive, Radio-Aware, Low-Cost (CORAL) Research Platform," *Proc. IEEE DySPAN*, May 2010.
- [10] Spectrum Bridge, Authorized Shared Access (ASA), <http://www.spectrumbridge.com>.

- [11] H. Harada et al., "A Software Defined Cognitive Radio System: Cognitive Wireless Clouds," *Proc. IEEE GLOBECOM*, Washington, DC, Nov. 2007.
- [12] WARP Repository, <http://warp.rice.edu/trac>.
- [13] HOPERF, HM-TR Series UHF Wireless Transparent Data Transceiver, HOPERF Co., June 2008.
- [14] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," *Commun. ACM*, vol. 51, no. 1, Jan. 2008, pp. 107–13.
- [15] C.-H. Ko, D.-H. T. Huang and S.-H. Wu, "Cooperative Spectrum Sensing in TV White Spaces: When Cognitive Radio Meets Cloud," *Proc. IEEE INFOCOM*, Shanghai, China, Apr. 2011.

## BIOGRAPHIES

SAU-HSUAN WU (sauhsuan@cm.nctu.edu.tw) received B.S. and M.S. degrees from National Cheng Kung University, Taiwan in 1990 and 1993, respectively, both in engineering science. From 1993 to 1995 he served in the Army of Taiwan, and from 1995 to 1999 he worked as a circuit and system design engineer in Taiwan. In 2003 he received a Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles. From 2003 to 2004 he was a postdoctoral research fellow at the Department of Electrical Engineering, University of Southern California. From 2004 to 2005 he served as a technical consultant for Winbond Electronics Corporation America, developing wireless MIMO-OFDM products. Since 2005 he has been an assistant professor at the Department of Electrical and Computer Engineering, National Chiao Tung University (NCTU), Taiwan. His research interests lie in the application of signal processing theory to wireless communication systems.

HSI-LU CHAO (hlchao@cs.nctu.edu.tw) received her Ph.D. degree in electrical engineering from National Taiwan University in 2004. Since August 2004 she has been an assistant professor, Department of Computer Science, NCTU. Her research interests lie in the area of design and performance analysis of wireless networks, including MAC and resource management in cognitive radio networks, millimeter-wave personal area networks, and LTE/LTE-A networks.

CHUN-HSIEN KO (junxian.cm99g@g2.nctu.edu.tw) received his B.S. degree from NCTU in 2010. Since September 2010, he has been a Ph.D. student in the Institute of Communications Engineering, NCTU. His research interests include machine learning and a parallel framework for cloud computing.

SHANG-RU MO (maxmo77715.cs99g@nctu.edu.tw) is a Master's student in the Institute of Computer Science and Engineering, NCTU. His research interests include cloud computing and resource allocation in cognitive radio networks.

CHUNG-TING JIANG (tim750811.cm98g@nctu.edu.tw) received his B.S. degree in Communication Engineering from NCTU in 2009. Since September 2009 he has been a Ph.D. student in the Institute of Communications Engineering, NCTU. His research interests include LDPC coding and reconfigurable radio.

TZUNG-LIN LI (ivljyt.cs99g@nctu.edu.tw) received his B.S. degree from National Central University (NCU), Taiwan, in 2010. He is now a master student in the Institute of Computer Science and Engineering at NCTU. His research focuses on the design and implementation of control channels for cognitive radio networks.

CHUNG-CHIEH CHENG [StM] (james770329.cm99g@nctu.edu.tw) received his B.S. degree in communication engineering from NCTU in 2010. He is currently a Master's student in Institute of Communications Engineering, NCTU, Taiwan. His research interests lie in the embedded system design and power management for cognitive radio networks.

CHIAU-FENG LIANG (streamaway.cs99g@nctu.edu.tw) received his B.S. degree in computer science and information engineering from NCU in 2010. He is now a Master's student in the Institute of Network Engineering, NCTU. He research focuses on the design of scheduling algorithms for cognitive radio networks.