OS-MAC: An Efficient MAC Protocol for Spectrum-Agile Wireless Networks

Bechir Hamdaoui, *Member*, *IEEE*, and Kang G. Shin, *Fellow*, *IEEE*

Abstract—Wireless networks and devices have rapidly been gaining popularity over their wired counterparts. This popularity, in turn, has been generating an explosive and ever-increasing demand for, and hence creating a shortage of, the radio spectrum. The reason for this foreseen spectrum shortage is reported not to be the scarcity of the radio spectrum but the inefficiency of current spectrum access methods, thus leaving spectrum opportunities along both the time and frequency dimensions that wireless devices can exploit. Fortunately, recent technological advances have made it possible to build software-defined radios (SDRs), which, unlike traditional radios, can switch from one frequency band to another at little or no cost. We propose a MAC protocol, called *Opportunistic Spectrum MAC* (OS-MAC), for wireless networks equipped with cognitive radios like SDRs. OS-MAC 1) adaptively and dynamically seeks and exploits opportunities in both licensed and unlicensed spectra and along both the time and frequency dimensions, 2) accesses and shares spectrum among different unlicensed and licensed users, and 3) coordinates with other unlicensed users for better spectrum utilization. Using extensive simulation, OS-MAC is shown to be far more effective than current access protocols from both the network's and the user's perspectives. By comparing its performance with an Ideal-MAC protocol, OS-MAC is also shown to not only outperform current access protocols, but also achieve performance very close to that obtainable under the Ideal-MAC protocol.

Index Terms—Spectrum agility, opportunistic MAC protocols, software-defined radios (SDRs), cognitive wireless networks.

◆

1 INTRODUCTION

The demand for radio spectrum has rapidly been increasing for several reasons. First, wireless networks and devices are rapidly gaining popularity over their wired counterparts mainly due to their low cost and convenience of use, which, in turn, has increased the demand for spectrum. Second, wireless applications are increasing in number, size, and complexity, thereby requiring more bandwidth and, hence, more demand for spectrum. Finally, advances in wireless technology have enhanced the quality of existing applications and created new wireless services, which also increases the demand for spectrum. For example, although technological advances in cellular networks created 3G that enabled high-speed data rates, they also contributed to higher consumer demand: Consumers now want to receive not only the traditional voice service but also Internet data services via their handheld devices. In contrast, the spectrum supply has not been keeping up with the spectrum demand. This expected shortage in spectrum supply has prompted both industry and federal agencies to explore new ways of making efficient use of the spectrum.

In November 2002, the Federal Communications Commission (FCC) established the Spectrum Policy Task Force (SPTF) to identify possible changes in the current spectrum allocation policies that will increase its overall public benefits. The US Defense Advanced Research Projects Agency (DARPA) also created the so-called Next-Generation (XG) Program that aims at developing a new generation of access technology [1] that will more efficiently use the spectrum. Industry organizations such as the MITRE Corporation [2], [3] and the IEEE 802.22 Working Group [4], [5] are also working on technologies and standards necessary for providing wireless devices with the capability of adaptive and dynamic spectrum access and sharing. Changing policies to allow dynamic spectrum allotment and developing techniques to enable opportunistic spectrum access are two major challenging issues that need to be resolved for the efficient use of limited and precious spectrum. Since the former issue falls within the domain of policy makers and regulatory bodies, we will focus on the latter issue.

Preliminary studies [6], [7], [8] indicate that the spectrum shortage problem is not so much due to the scarcity of the radio spectrum but due to the inefficiency of current spectrum allocation methods. For instance, from the actual measurements of spectrum use in several major US cities during various periods in July 2002, it was observed that many portions of the radio spectrum below 1 GHz are not in use for significant periods of time [7]. Likewise, measurements taken during the period between January 2004 and August 2005 show that only about 5 percent of the spectrum is actually in use in the band below 3 GHz at any location in the US and at any time [8]. These indicate the availability of ample spectrum opportunities—often also referred to as "white spaces"—for wireless devices to exploit along both the time dimension (resulting from variability of spectrum usage over time) and the frequency dimension (resulting from variability of spectrum usage over different frequency bands).

Due mainly to technology limitations, spectrum has traditionally been "statically" licensed and assigned in blocks via frequency division. However, technological

---

- *B. Hamdaoui is with the Electrical Engineering and Computer Science Department, Oregon State University, 1148 Kelley Engineering Center, Corvallis, OR 97331. E-mail: hamdaoui@eecs.orst.edu.*
- *K.G. Shin is with the Electrical Engineering and Computer Science Department, University of Michigan, 2260 Hayward Ave, Ann Arbor, MI 48109. E-mail: kgshin@eecs.umich.edu.*

Manuscript received 15 Aug. 2006; revised 13 May 2007; accepted 15 Aug. 2007; published online 12 Sept. 2007.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0214-0806.
Digital Object Identifier no. 10.1109/TMC.2007.70758.

advances enabled Software-Defined Radios (SDRs), unlike traditional radios, to switch from one frequency band to another at minimal cost. SDRs are expected to be a key component of future wireless systems and applications and will empower wireless devices with the capability of dynamically accessing the entire frequency band. This paper proposes a new protocol for cognitive wireless networks that empowers SDR-based wireless devices with the following capabilities:

- adaptively and dynamically seeking and exploiting opportunities in both licensed and unlicensed spectra and along both the time and frequency dimensions,
- accessing and sharing spectrum among different unlicensed and licensed users, and
- coordinating with other unlicensed users for better spectrum utilization.

The effectiveness of OS-MAC is extensively evaluated using an ns2-based simulation. The performance of OS-MAC is compared with 1) that of existing spectrum-access methods and 2) that of an Ideal-MAC protocol, demonstrating that OS-MAC is far more effective than current access protocols from both the network's and the user's perspectives. Moreover, OS-MAC is shown to not only outperform current access protocols but also achieve performances that are very close to those achievable under an Ideal-MAC protocol.

The rest of this paper is organized as follows: In Section 2, we discuss the requirements for achieving spectrum agility. Section 3 describes the proposed OS-MAC protocol. Section 4 evaluates the effectiveness of OS-MAC using an ns2-based simulation. Section 5 discusses the applicability/implementation of OS-MAC. The related work is discussed in Section 6. Finally, we conclude this paper in Section 7.

2 SPECTRUM AGILITY: A DESIGN GUIDELINE

In order to design MAC protocols that can fully exploit the available spectrum in both licensed and unlicensed frequency bands, one must first understand spectrum allocation policies and recognize their access limitations.

R1. Spectrum Regulations. At present, the FCC statically divides radio spectrum into frequency bands and assigns them to users according to one of the three models [6], which, for simplicity, we classify into two types. The first type is to allocate frequency bands to licensees, referred to as *Primary* Users (PUs), who have exclusive and flexible rights to use their assigned spectrum. PUs are also protected against interference when using their assigned spectrum. The second type is to allow other users, referred to as *Secondary* Users (SUs), to share the remaining spectrum (that is, unlicensed spectrum) in a nonexclusive manner. Unlike PUs, SUs have neither rights to nor guarantees of interference protection. To improve spectrum efficiency, regulatory bodies such as the FCC need to revise their spectrum-leasing policies and/or pursue market regulations that encourage licensees to provide SUs with opportunistic access to their spectrum bands.

R2. Interference Avoidance. Since PUs have exclusive access rights to their allocated spectrum bands, SUs can *opportunistically* use licensed spectrum only if their signals do not cause interference to PUs. That is, upon the detection of the presence of PUs, SUs must immediately vacate the channel if they happen to be using the licensed spectrum band. Note that detection mechanisms are beyond the scope of this paper. Readers may refer to [9], [10], [11], and [12] for the methods that SUs can use to detect the presence of PUs. Hence, access methods for promoting spectrum efficiency must enable SUs to suppress their signals or immediately vacate the licensed spectrum upon the detection of PUs.

R3. Spectrum Access Sharing. Since different SUs may simultaneously seek spectrum opportunities, multiple different SUs can simultaneously move to and use the same spectrum band. Thus, opportunistic spectrum access methods must support the coexistence of multiple SUs in the same spectrum band.

R4. Spectrum Access Efficiency. The SPTF identified three forms of efficiency—spectrum, technical, and economical—to improve. From the MAC's perspective, it is the first form of efficiency that needs to be achieved. Hence, spectrum access methods must provide SUs with collaborative capabilities for spectrum efficiency.

The proposed protocol OS-MAC is designed in accordance with the four design requirements above.

3 OPPORTUNISTIC SPECTRUM MEDIA ACCESS CONTROL PROTOCOL

3.1 Assumptions and Notation

OS-MAC is developed under the following assumptions:

- The available radio spectrum is equally divided into N nonoverlapping data channels[1] (DCs) and one common control channel (CC). We assume that the spectrum division into DCs and CC is done by a third authoritative party (for example, the FCC) and that all SUs have prior knowledge of such division. Each DC is associated with a number of PUs that have exclusive and flexible use and access rights to use it. PUs can use their own DC at any time.
- We use the notion of an SU Group (SUG) to represent a set of users who want to communicate with each other: an SUG may consist of two or more SUs. At any time, only one member in an SUG can transmit information at a time, and the rest in the same group will receive it (this is akin to one member talking and the others listening in a group discussion). There may be multiple SUGs in the network, all of which simultaneously seek spectrum opportunities on all DCs to establish communications. We will hence call these types of communication *sessions*. SUGs can seek and use any DC, as long as the DC is not being used by its PUs. That is, upon the detection of the presence or the return of PUs, SUs must immediately vacate the DC.

  The rationale behind the above SUG model is to design a MAC that supports not only the traditional one-to-one communication sessions, but also the new emerging many-to-one communication sessions such as teleconferencing. Note that a pair of communicating users can be viewed as a special case of an SUG with only two members.

---

[1] From now on, we will use the term "channel" to refer to "spectrum band."

- We assume that an SU can directly communicate with any other SU when tuned to the same channel. Under this assumption, mobility is not an issue, provided that users stay connected during their communication. However, if the condition of the wireless spectrum worsens due to mobility or any other factors, the adaptive feature of the proposed protocol allows SUs to seek and switch to other better quality spectrum bands.
- Each SU is equipped with a single half-duplex transceiver to transmit or receive on one channel at any given time.

3.2 General Description of Opportunistic Spectrum Media Access Control

All interchannel control frames are communicated via the CC, whereas DCs are used for communicating all data frames and all intrachannel control frames. Each DC will always have one Delegate SU (DSU) appointed among those SUs currently using it. All DSUs (one from each DC) periodically switch to CC to inform each other of the traffic loads experienced on their DCs. After learning of the conditions of all DCs, each DSU returns to its original DC and informs all SUGs currently using that DC about the traffic conditions of all the other DCs. Based on this information, each SUG selects and then switches to the "best" DC for data communication until the end of the current period. Although DSUs are in the CC, informing each other of their channel conditions, all other SUs continue using their DCs for normal data communications.

We propose that all SUs within the same DC use the IEEE 802.11 DCF access mode (without RTS/CTS) [13] (see the Appendix for a summary of the IEEE 802.11). However, since an SU sender may send information to multiple receivers, only one receiver will acknowledge the receipt of a packet as follows: Upon receiving a packet, each receiver sets a random backoff timer. If the receiver sees an ACK (from a different receiver) prior to the expiration of its timer, then it cancels the timer. If its timer expires before seeing any ACK, then the receiver sends an ACK. Recall that all members belonging to the same SUG are assumed to all hear each other. Therefore, having only one receiver acknowledge the receipt of a packet will suffice. Here, ACKs are used to handle delivery failures caused by collisions. Our MAC protocol also provides the option of turning off the acknowledgment mechanism; that is, no ACKs are sent back to the sender. This can be used to support sessions whose communication quality is not too sensitive to packet losses.

In summary, OS-MAC divides time into periods, each of which is called the *Opportunistic Spectrum Period* (OSP) and consists of three consecutive phases: Select, Delegate, and Update. The lengths, in time slots, of these three phases are denoted by `SelWin`, `DelWin`, and `UpWin`, respectively. Events occurring during each of these phases are briefly described as follows:

- **Select Phase**. Each SUG selects the "best" DC and uses it for communication until the end of the current OSP.
- **Delegate Phase**. On each DC, a DSU is appointed among those currently using the channel to represent the group during the Update Phase.
- **Update Phase**. All DSUs switch to the CC to update each other about their channel conditions, whereas all non-DSUs continue communicating on their DCs.

3.3 Details of Opportunistic Spectrum Media Access Control

Under OS-MAC, each SU in the network will be in one of the following phases at any given time:

1. **Network Initialization Phase**. If an SU is not involved in any communication, it will tune its transceiver to the CC. The SU will keep listening to the CC, unless it does the following:
    - It decides to establish a new session (and, hence, forms a new SUG). In this case, it moves to phase 2.
    - It receives a JoinRequest control frame from another SU that requests it to join/form an SUG. In this case, it replies with a JoinReply control frame and switches to the DC indicated in the received JoinRequest frame. It then moves to phase 3 to start communication with others in the group.
    - It decides to join an existing SUG/session. In this case, it scans all the DCs until it detects its desired SUG. Here, we assume that the SU has prior knowledge about its desired SUG, including its presence and ID. Upon detecting the desired SUG, the SU moves to phase 3.
2. **Session Initialization Phase**. If an SU wants to establish a new session, it will set its SessionInitialization timer to[2] $\texttt{InitWin} = (\texttt{MaxSelWin} + \texttt{DelWin} + 2 \times \texttt{UpWin})$ and keep listening to CC, unless any of the following occurs:
    - The SessionInitialization timer expires prior to receiving any UpdateCC control frame. UpdateCC frames are periodically sent on the CC by DSUs to inform each other of channel conditions during the Update Phase (more in phase 4). In this case, the SU (the only SU currently active in the network) will initialize its OS-MAC Parameter Set (to be defined later), select a random DC, inform its group members about the chosen DC, and switch to that channel for data communication (moving to phase 3).
    - The SU receives UpdateCC frames prior to the expiration of its SessionInitialization timer. In this case, the SU will update its OS-MAC Parameter Set to those indicated by the UpdateCC frames, select its "best" DC (via the Select Mechanism), inform its group members of the chosen DC, and switch to that DC for data communication (moving to phase 3).
3. **Data Communication Phase**. During the last `UpWin` time slots of each OSP, DSUs will switch to the CC to invoke the Update Mechanism (moving to phase 4). When DSUs switch to the CC, all other SUs will

---

[2] This timer is decremented by one every time slot and expires when it reaches 0. `InitWin`, `MaxSelWin`, `DelWin`, and `UpWin` are system design parameters that will be defined later.

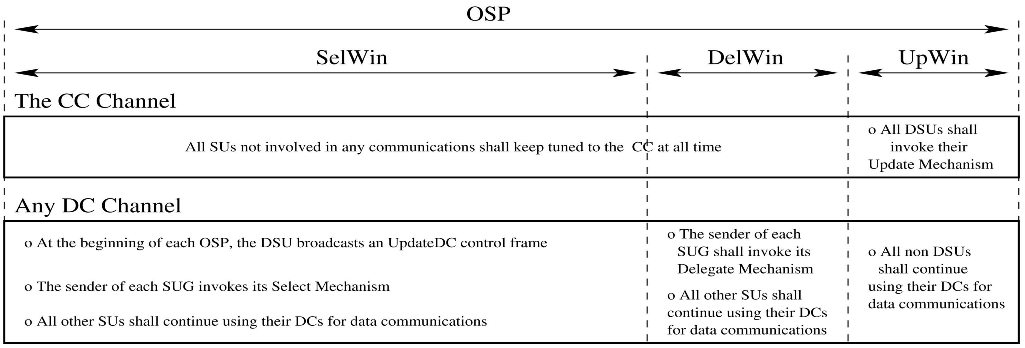


Fig. 1. OS-MAC.

continue using their DCs for data communications. When the Update Mechanism ends, each DSU will switch back to its DC to inform all the SUs (currently using the same DC) of the conditions of all other DCs that it just learned via the Update Mechanism. To convey this information, the DSU will broadcast an UpdateDC control frame upon switching back to its DC. This control frame will then signal the beginning of a new OSP by indicating all its parameters (see Sections 3.4 and 3.8 for details). Upon receiving the UpdateDC frame, the sender of each SUG will invoke its Select Mechanism (moving to phase 5) to select the "best" DC. Depending on which DC is selected by the Select Mechanism, all members of an SUG may switch to a new DC or remain in the same DC. In either case, SUGs will use the chosen DC to communicate until the end of this new OSP. At the end of the Select Phase, SUs will invoke the Delegate Mechanism (moving to phase 6) to appoint the DSUs that will represent DCs during the next OSP.

4. **Update Phase**. During the last `UpWin` time slots of each OSP, each DSU will be tuned to the CC to invoke its Update Mechanism. This mechanism consists of having each DSU send an UpdateCC control frame to inform other DSUs of its DC's traffic condition. By the end of this window period, all DSUs will tune their transceivers back to their original DCs and return to phase 3.
5. **Select Phase**. After returning from the CC to their DCs (that is, after phase 4), DSUs will immediately broadcast an UpdateDC control frame that informs all SUs of the current traffic conditions of all DCs. All control frames communicated on the DCs will have shorter DIFS periods to give them access priority over the channel. OS-MAC uses PIFS $(= \mathrm{SIFS} + \texttt{TimeSlot})$ as the time to wait for all control frames. Upon receiving this information, the sender of each SUG will invoke its Select Mechanism. Based on information in the UpdateDC frame, this mechanism allows senders of all SUGs to 1) choose the "best" DCs that they will use next and 2) inform all their SUs members of the selected DC. All members of the SUG will immediately switch to the selected DC for data communication.
6. **Delegate Phase**. At the beginning of each Delegate Phase and during a period of `DelWin` time slots, SUs on each DC will invoke their Delegate Mechanism to appoint their DSU that will represent their DC during the next OSP.

Fig. 1 summarizes all the above phases of the protocol.

3.4 Opportunistic Spectrum Media Access Control Parameters

Each SU always maintains and periodically updates a data structure, called the OS-MAC Parameter Set, that consists of five elements:

- $\varphi()$. This is a vector with as many elements as the number of DCs, where an element corresponding to a DC is responsible for holding and keeping track of the *access-time share*,[3] defined as the ratio of the time during which the SU possesses the DC during the Select Phase to that of the total length of the Select Phase, that SUs using the DC are currently receiving. The element of an SU's vector that corresponds to the DC that is used by the SU itself is updated by the SU during the Select Phase by measuring the fraction of time that the SU has access to the DC. The other elements of the vector are periodically updated during the Update Phase on the CC if the SU is the delegate of the DC. Otherwise (that is, if the SU is not a delegate), they are updated upon receiving an UpdateDC from the DSU. As we describe next, updating the `SelWin` parameter and determining the new "best" DC are both based on the information contained in $\varphi()$.
- `SelWin`. This is the length (in time slots) of the current Select Phase. This window determines how long SUs should wait before seeking better DCs by examining the conditions of other DCs. Note that the window is adjusted to the conditions of the channels. It is adaptively calculated based on the vector $\varphi()$ as

$$\begin{aligned}\texttt{SelWin} = &-4(\texttt{MaxSelWin} - \texttt{MinSelWin})\\ &\times \mathrm{var}[\varphi()] + \texttt{MaxSelWin},\end{aligned} \tag{1}$$

where `MaxSelWin` (upper bound) and `MinSelWin` (lower bound) are two design parameters, and $\mathrm{var}[\varphi()]$ is the variance of the $\varphi()$ vector. Fig. 2

---

[3] Note that because OS-MAC assumes that all DCs support the same data rate, the "access-time share" metric can also be viewed as a way of measuring the "obtainable throughput share." The "throughput share" can be computed as the "access-time share" multiplied by the bandwidth of the DC. See Section 4.7 to see how OS-MAC is also suitable for the case where all or some of the DCs do not support the same data rate.

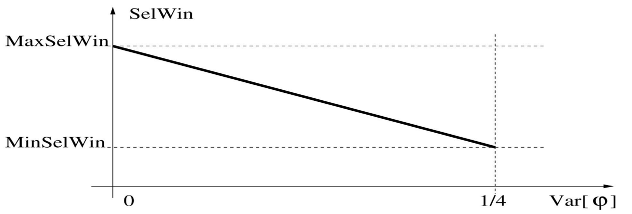


Fig. 2. Adaptation of `SelWin`.

shows `SelWin` as a function of $\mathrm{var}[\varphi()]$. Note that $\mathrm{var}[\varphi()]$ varies between 0 and 1/4 because each element of the $\varphi()$ vector is a ratio that can only be between 0 and 1. Further clarifications regarding (1) are provided in Section 3.1.3.

- `DelWin`. This window need not be adaptive (see Delegate Mechanism). It is a design parameter that should be large enough to allow at least one successful transmission. This window is typically much smaller than `SelWin`.
- `UpWin`. This window need not be adaptive (see Update Mechanism). It should be large enough to allow at least N successful UpdateCC control frames. This window is typically much smaller than `SelWin`.
- `PeriodStartTime`. This is the start time of the next OSP. It is the old value of `PeriodStartTime` plus the values of the three windows `SelWin`, `DelWin`, and `UpWin`.

3.5 Select Mechanism

One major challenge in designing OS-MAC is how the phenomenon of "synchronizing behaviors" can be resolved. Due to their adaptive nature of locating and switching to the best spectrum band (that is, less loaded, less noisy, and so forth), several SUGs may end up all switching to the same band, thereby rendering it the worst. This undesired phenomenon leads not only to a lesser achievable per-SUG throughput, but also to an overall degradation of the spectrum utilization. The following Select Mechanism is designed to avoid this.

Upon receiving an UpdateDC control frame that contains an updated $\varphi()$ vector, a sender S of a given SUG currently using DC i will select a new DC as follows:

Let $\overline{\varphi} = \frac{N}{\sum_{l=1}^{l=N} \frac{1}{\varphi(l)}}$ and $\mathcal{A} = \{DC\, j : \varphi(j) > \overline{\varphi}, j = 1, 2, \ldots, N\}$.

1. If $\varphi(i) > \overline{\varphi}$, S will remain on the same DC i.
2. Else
   - with probability $\frac{\varphi(i)}{\overline{\varphi}}$, S will remain on the same DC i, and
   - with probability $(1 - \frac{\varphi(i)}{\overline{\varphi}}) \frac{\varphi_r(j)}{\sum_{k \in \mathcal{A}} \varphi_r(k)}$, S will select DC $j \in \mathcal{A}$ where $\varphi_r(k) = \frac{\varphi(k) - \overline{\varphi}}{\varphi(k)}, \forall k \in \mathcal{A}$.

Note that the above selection algorithm is executed by the sender of each SUG only. Once decided, the sender will immediately inform each of its members about the chosen DC via a JoinRequest control frame. Once informed, all members will switch to the chosen DC. Note that an SUG may stay on the same DC during the next OSP since the outcome of the Select Mechanism may be the same DC. In such a case, no JoinRequest frame is sent.

There are several points that require mentioning regarding our Select Mechanism. First, since at any given time, only one member of each SUG can be transmitting, then the number of SUs in an SUG does not affect the SUG's share of bandwidth. In other words, an SUG's share of the bandwidth depends on the total number of SUGs currently using the system and not on the total number of SUs. Second, note that the proposed Select Mechanism is stable in that it prevents unnecessary DC switches by neither allowing SUGs whose access-time shares are higher than the average to switch their DCs nor allowing those with shares below the average to switch to other DCs whose shares are also below the average. Finally, recall that our protocol strives to ensure that each SU receives an equal share of the available bandwidth by adaptively switching to bands with better bandwidth shares. Hence, our fairness criterion is to assure that each user receives the same throughput. Although bandwidth fairness within each DC is ensured via the IEEE 802.11 access mechanism, bandwidth fairness across all DCs is ensured via the Select Mechanism of our MAC protocol. In the following, we formally state and prove this last feature.

Proposition 1. *For every DC j, $E[\varphi(j)] = \overline{\varphi}$.*

Proof. Let n_j^0 and n_j denote the number of SUGs occupying DC j before and after invoking the Select Mechanism, respectively. Let us arrange the set of N channels as $\{1, 2, \ldots, l, l+1, \ldots, N\}$ such that $\varphi(j) \le \overline{\varphi} \Leftrightarrow j \le l$. Without loss of generality, we assume that $\varphi(j)$ is inversely proportional to n_j^0. Let $\varphi(j) = \frac{1}{n_j^0}, \forall j$. Let X_{ij} denote the random variable representing the number of SUGs currently belonging to DC i that decided to switch to DC j after invoking the Select Mechanism. Note that $X_{ij} \sim B(n_j^0, p_{ij})$, where $p_{ij} = (1 - \frac{\varphi(i)}{\overline{\varphi}}) \frac{\varphi_r(j)}{\sum_{k \in \mathcal{A}} \varphi_r(k)}$, and hence, $E[X_{ij}] = n_i^0 p_{ij}$. We will show that $E[n_j] = \frac{1}{\overline{\varphi}}$ for all j. Let $a = \frac{1}{N} \sum_{p=1}^{p=N} n_j^0$. There are two cases to consider:

Case 1. $j = l+1, l+2, \ldots, N$.

$$\begin{aligned} E[n_j] &= n_j^0 + \sum_{i=1}^{i=l} E[X_{ij}] = n_j^0 + \sum_{i=1}^{i=l} n_i^0 p_{ij} \\ &= n_j^0 + \frac{(a - n_j^0)}{\sum_{p=l+1}^{N} (a - n_p^0)} \sum_{i=1}^{i=l} (n_i^0 - a) = a, \end{aligned}$$

since $\sum_{i=1}^{l} (n_i^0 - a) = \sum_{i=l+1}^{N} (a - n_i^0)$.

Case 2. $j = 1, 2, \ldots, l$. $E[n_j] = \frac{\varphi(j)}{\overline{\varphi}} n_j^0 = a$. □

3.6 Delegate Mechanism

The purpose of this mechanism is to appoint a new DSU that will represent the DC during the next Update Phase. The idea is simple. When the Delegate Phase begins, all SUGs belonging to a given DC continue competing for the DC through the 802.11 random access scheme. Recall that, for each successful transmission, an ACK will be sent back to the sender. Under the single-hop assumption, these ACKs will be heard by all SUs of the DC. The first sender that successfully delivers a packet during the Delegate Phase is automatically appointed as the new DSU. Therefore, upon receiving an ACK notifying a successful reception, the sender considers itself the new DSU. Any other SU that is using the same DC will also hear the ACK and hence would know that someone else is appointed as the DSU. Once appointed, the DSU will act as the delegate until it sends an **UpdateDC** at the beginning of the next Select Phase. Once appointed as a delegate, a DSU should not quit until it sends the **UpdateDC**, even when its session ends earlier. This is not an issue since the length of the combined Delegate and Update Phases is on the order of seconds, whereas the length of sessions that we consider is on the order of a dozen of minutes.

Note that the Delegate Mechanism is 1) simple and 2) incurs no overhead. It is simple because it exploits the already-existing ACK mechanism. The mechanism is distributed and does not require any extra message exchange, hence incurring no bandwidth overhead.

3.7 Update Mechanism

The length of the Update Phase (that is, `UpWin` time slots) is divided into N identical intervals. Upon switching to the CC, a DSU representing channel j will broadcast its **UpdateCC** control frame on the $j\text{th}$ interval. During the `UpWin` time slots, upon receiving an **UpdateCC** frame, each SU will update its 1) TSF timer to that indicated in the `Timestamp` field and 2) the $\varphi()$ structure to that indicated by the $\varphi(n)$ field of the frame received from channel n.

3.8 Control Frame Formats

3.8.1 UpdateCC Control Frame

During the Update Phase, each DSU will broadcast an **UpdateCC** control frame to inform other DSUs of the condition of the DC that it represents. Each **UpdateCC** frame will contain the following fields:

- `Timestamp`: Indicates the current time and is necessary for time synchronization.
- `PeriodStartTime`: Indicates the start time of the new OSP. This information is needed for SUs that just joined the network.
- $\varphi(n)$: The time share in accessing the channel experienced in DC n.

3.8.2 UpdateDC Control Frame

As soon as the DSU switches back to its DC (after invoking the Update Mechanism), it will broadcast an **UpdateDC** control frame on the DC. The purpose of this frame is to inform all other SUs of the channel-access-time shares of the other DCs. This frame also signals the debut of a new OSP. An **UpdateDC** frame will contain the following fields:

- `Timestamp`: This indicates the current time and is needed for time synchronization.
- `SelWin`: The length of the Select Phase.
- `DelWin`: The length of the Delegate Phase.
- `UpWin`: The length of the Update Phase.
- $\varphi()$: This vector contains the channel access times of all DCs (see Section 3.4 for details).

3.8.3 JoinRequest Control Frame

Whenever an SUG decides to switch to a new (better) DC, the delegate of the group will send a **JoinRequest** frame to all its members, informing them of that DC. The best DC is determined via the Select Mechanism, as described in Section 3.5. As mentioned earlier, there are two scenarios during which a **JoinRequest** may be sent: at the session initialization phase or upon receiving an **UpdateDC**. This frame will contain the following fields:

- `SrcAddr`: The source address of the sender.
- `DstAddr[]`: An array containing the destination addresses of all the members of the SUG.
- `TargetDC`: The best DC to use during the current OSP.

3.8.4 JoinReply Control Frame

This frame acts as an ACK to a **JoinRequest**. Similar to acknowledging data packets, only one receiver will acknowledge the receipt of a **JoinRequest**. **JoinReply** will contain the destination address of the sender `DstAddr`.

3.9 Features of OS-MAC

Having detailed OS-MAC, we would now like to reiterate and provide the intuitions behind its features.

3.9.1 Efficient Usage of Spectrum

Since only one SU from each DC switches to the CC during the Update Phases while the other SUs continue using their DCs, there is no wastage of spectrum opportunities. Because OS-MAC uses the contention-based IEEE 802.11 access method, SUs will always probe and use all available bandwidth, independent of how many SUs are actually using the DC. In fact, each SU using a DC will, on the average, receive an access-time share that is inversely proportional to the total number of SUs currently using the same DC.

3.9.2 Negligible Control Overhead

The control overhead (in terms of spectrum wastage) associated with OS-MAC is primarily due to the Update Phase during which DSUs leave their DCs and switch to the CC to exchange control frames. Obviously, no spectrum would be wasted if each DC has more than one SUG using it such that when DSU switches to the CC, the other SUGs continue using the DCs. Only when the spectrum is lightly loaded can OS-MAC incur a negligible overhead in terms of spectrum wastage. The overhead is negligible because the length of the Update Phase (in the order of a second) is very small as compared to that of the period (OSP) (on the order of a dozen of minutes). Besides, here, we are not dealing with lightly loaded networks. In fact, when networks are lightly loaded, there is no need for spectrum-agile MACs in the first place.

3.9.3 Adaptability to Spectrum Condition

At first glance, one can note that, the shorter the OSPs, the more balanced the traffic load across all DCs and the higher the spectrum wastage as the Update Phases occur more frequently. Although the former maximizes the spectrum efficiency, the latter does the opposite. OS-MAC deals with these two conflicting objectives by adaptively adjusting the length of OSPs to current network traffic loads.[4]

---

[4] Recall that the length of OSP is the sum of `SelWin`, `SelWin`, and `UpWin`. Because both `DelWin` and `UpWin` are fixed, the adaptation of OSP comes from that of `SelWin`, as shown in (1).

The philosophy behind OS-MAC's adaptation of the lengths of OSPs to the traffic load is simply to strive in ensuring that all serviced sessions receive an equal access-time share (and, hence, an equal bandwidth share) of unused spectrum, regardless of which DC they use. Achieving an equal access-time share across all sessions 1) guarantees balanced loads across all DCs and, hence, efficient overall spectrum utilization and 2) maximizes the per-session quality of communication. OS-MAC relies on the variance of $\varphi()$ (the access-time shares received on all DCs) to determine whether sessions receive an equal share or not and accordingly adjust the lengths of OSPs. That is, an increase in the variance is interpreted by OS-MAC as an indication of unbalanced traffic load over DCs. As a result, OS-MAC reduces the length of OSP so that sessions with small shares can switch to DCs whose received shares are higher. On the other hand, when OS-MAC detects small variances, it increases the length of OSP, as this implies that sessions are likely to receive the same share independent of the DC that they use. Hence, there is no need for them to seek/change their DCs any time soon (see (1) for a formal definition of `SelWin`).

Now, let us consider the wastage of spectrum due to the Update Phases under this adaptation. When the traffic load is light, the variance of $\varphi()$ is small, and hence, the lengths of OSPs are large. This is true regardless of whether the load is balanced. Therefore, under light loads, the spectrum wastage due to the Update Phases is minimal because these phases infrequently occur. On the other hand, as discussed in Section 3.1.2, under medium to high traffic loads and independent of the lengths of OSPs, OS-MAC incurs no spectrum wastage.

In essence, this adaptation technique of OS-MAC maximizes the overall spectrum efficiency at very little or no control overhead. This is confirmed via our simulation results, as presented in Section 4.

3.9.4 Collaborative Spectrum Sharing

OS-MAC uses the CC as a means for all SUs, seeking and using available spectrum opportunistically to collaborate for better performance at both the user and the network levels with minimum overhead. OS-MAC assures collaboration via its Update Mechanism by which representatives from each DC periodically switch to the CC to update each other with channel conditions. Because only one SU from each DC switches to CC, whereas the other SUs continue using their DCs, OS-MAC maintains coordination among users at no or little spectrum wastage. As we will discuss in Section 5, the bandwidth allocated to the CC should be just large enough to support interchannel control traffic.

3.9.5 Avoidance of Synchronizing Behaviors

Another important feature of OS-MAC lies with its Select Mechanism. This mechanism allows SUGs to seek and use the spectrum opportunistically by adaptively and dynamically switching to less crowded DCs. Although the Select Mechanism is formally described in Section 3.5, we provide here insights and intuitions on how it works. First, under OS-MAC, only SUGs whose received access-time shares are below the average should seek and switch to DCs with higher received shares. Others should remain on their DCs. To avoid synchronizing behaviors in which all or many SUGs with low shares switch to the DC whose received share is the highest, OS-MAC uses a probabilistic selection approach. That is, with a given probability, some of those SUGs whose received shares are below the average will remain on their DCs or possibly choose a DC with a higher received share than the average but not the highest. Hence, without incurring any synchronizing behaviors, the Select Mechanism statistically assures that all sessions receive "almost" equal access-time shares by periodically having some (not all) sessions seek opportunistic DCs (see Proposition 1 for a formal proof). Moreover, the Select Mechanism prevents unnecessary DC switches from occurring by neither allowing SUGs whose shares are higher than the average to switch their DCs nor allowing those with shares below the average to switch to other DCs whose shares are also below the average.

4 PERFORMANCE EVALUATION

The effectiveness of OS-MAC is extensively evaluated via an ns2-based simulation. OS-MAC is comparatively evaluated along with R-MAC, MC-MAC, and Ideal-MAC.[5] Each simulation run took 2 days and was repeated 10 times, each with a different seed. All reported results were averaged over 10 seeds.

4.1 Simulation Method and Parameter Setting

The spectrum is divided into N nonoverlapping DCs and one CC, each of which has a capacity of B bits per second (bps). Each DC is associated with a number of PUs that have exclusive right to access it. PUs may use their own DC at any time. We characterize each DC n with ON (busy) and OFF (idle) periods of exponentially distributed lengths with means $\lambda_{ON}(n)$ and $\lambda_{OFF}(n)$, respectively. The parameters $\lambda_{ON}(n)$ and $\lambda_{OFF}(n)$ are used to control the DC n's traffic load resulting from PUs. These parameters are also allowed to simulate cases where different DCs experience different loads. Let

$$\eta_p(n) = \frac{\eta_{ON}(n)}{\eta_{ON}(n) + \eta_{OFF}(n)},$$

and δ_P denote the PUs' average traffic load on DC n and the coefficient of variation of PUs' traffic loads across all the N DCs. In addition, let

$$\eta_P = \frac{1}{N}\sum_{n=1}^{N} \eta_P(n)$$

denote the PUs' average load on all DCs.

Along with PUs, SUs seek and use the spectrum left unused by PUs by forming groups, establishing sessions, and communicating on DCs. There are M SUGs in the network. Here, we assume that $M \gg N$ (otherwise, the problem becomes trivial). During the course of simulation, sessions are randomly generated by SUGs as follows: Each SUG generates sessions, each of size Z bytes selected from a uniform distribution with mean $\bar{Z}$ and coefficient of variation δ_Z. Between every two consecutive sessions, each SUG goes into an idle period also selected from a uniform distribution with mean $\bar{I}$ and coefficient of variation δ_I. The packet length is set to L bytes. Let

[5] R-MAC, MC-MAC, and Ideal-MAC are described in Section 4.2.

TABLE 1
Simulation Parameters

sym.	Description	Value
N	Number of available DCs	5
M	Number of Secondary User Groups	30
B	Capacity of each DC	1 Mbps
L	Length of packets	1250 Bytes
δ_I	Coef. of variation of idle periods	50%
δ_P	Coef. of variation of loads due to PUs	50%

$$\eta_S = \frac{M}{N}\frac{\bar{Z}}{\bar{Z}+\bar{I}B}$$

be the SUs' per-DC traffic load. Hence, DC n's traffic load due to all users is $\eta(n) = \eta_P(n) + \eta_S$. Note that the network parameter η constitutes an upper bound on the average per-DC achievable utilization.

We run simulations for different values of η_P and η_S. These values are controlled via the simulation parameters of PUs ($\lambda_{ON}(n)$ and $\lambda_{OFF}(n)$ for every DC n) and those of SUs ($\bar{Z}$ and $\bar{I}$). All measurements are taken in the same way for all three protocols: OS-MAC, R-MAC, and MC-MAC. All the other simulation parameters are fixed as indicated in Table 1.

We consider the performance of all three protocols for three network scenarios:

- $\eta_P(n) = 0$ on each DC n. PUs are not present.
- $\eta_P(n) = 30\%$ on each DC n. PUs are present, provided that they generate a total traffic load of 30 percent.
- $\eta_P(n) = 60\%$ on each DC n. PUs are present, provided that they generate a total traffic load of 60 percent.

Let us now elaborate on how the parameters `DelWin`, `UpWin`, `MinSelWin`, and `MaxSelWin` of our proposed OS-MAC are to be chosen. The length of `DelWin` must be large enough to permit for the successful delivery of at least one packet. Note that since the Delegate Phase incurs no extra control traffic overhead, the length of such a parameter is not so crucial to the performance of the protocol. We chose `DelWin` to be 5 seconds. The length of `UpWin` must also be large enough to permit for at least N successful UpdateCC control frames. Unlike `DelWin`, `UpWin` depends on the number of spectrum bands N. Hence, it is a design parameter. In our simulation, we set it to 1 s, which is long enough for 5 DCs ($N = 5$). As for the parameters `MinSelWin` and `MaxSelWin`, since we consider communication sessions of length of the order of a dozen minutes, we set `MinSelWin` and `MaxSelWin` to 5 and 15 minutes, respectively. These are also design parameters.

4.2 Random Media Access Control, Multichannel Media Access Control, and Ideal Media Access Control

OS-MAC is compared with the following MACs: R-MAC, MC-MAC, and Ideal-MAC.

4.2.1 Random Media Access Control

First, there is currently no commercial protocol or device that supports dynamic and adaptive multiband access to the spectrum. In the current technology (for example, IEEE 802.11), users in the unlicensed band can select and use one spectrum band among several available bands, but such a selection is statically done. Therefore, to compare our protocol with current access methods, we defined and introduced the Random-MAC (R-MAC) protocol to mimic current commercial multiband access methods. Like OS-MAC, R-MAC also uses a dedicated CC for interchannel control traffic while using DCs for data communications. R-MAC works as follows:

- When an SU wants to establish a session and hence form an SUG, it will randomly select one of the N DCs, inform all its members of the selected DC, and switch to that DC for immediate data communication.
- All members of an SUG will use only one DC during each session. That is, they are not allowed to switch DCs during a session. Upon the detection of PUs on their selected DC, all members on the channel will cease transmissions, as long as the DC is occupied by the PUs. Only when the DC is sensed idle again could the members resume transmission. Once their session ends, all members switch back to the CC.
- Like OS-MAC, multiple SUGs that selected the same DC will share the channel in accordance with carrier sense multiple access with collision avoidance (CSMA/CA), as specified in IEEE 802.11 [4].
- SUs that want to join ongoing sessions will scan the N DCs to detect the group that they intend to join.

4.2.2 MC-MAC

Like OS-MAC, MC-MAC [14],[6] which is a multichannel access protocol, uses a single half-duplex transceiver, a dedicated CC, and N DCs. Time is divided into beacon intervals. At the beginning of each interval, all SUs switch to the CC for a short period of time called ATIM window. During this window, source-destination pairs negotiate and select their "best" DCs to communicate their packets. Upon agreeing on a DC, the pair switches to it for DATA/ACK packet transmission until the end of the beacon interval. Each SU maintains a data structure, called Priority Channel List (PCL), holding information regarding the busyness of each DC. An entry of a node A's PCL corresponding to DC n will be in one of the three preference states at all time:

- *LOW*. n has been selected by a neighbor of node A to use during the current beacon interval.
- *MID*. No neighbor of node A has selected this n for use during the current beacon interval.
- *HIGH*. n has only been selected by node A (among its neighbors) for use during the current interval.

Negotiation and selection of DCs among SUs are done via a three-way handshake: ATIM-REQ, ATIM-ACK, and ATIM-RES. Before transmitting packets, the sender switches to the CC and waits for the ATIM window to send an ATIM-REQ message with PCL information to its receiver. After receiving the ATIM-REQ, the receiver selects the best channel as follows:

- If there is a HIGH-state DC in the receiver's PCL, this DC is selected.

---

[6] Readers may refer to [14] for more details.

- Else, if there is a HIGH-state DC in the sender's PCL, this DC is selected.
- Else, if there is a DC with the MID state in both PCLs, this DC is selected.
- Else, if there is a DC with MID in only one side, this DC is selected.
- Else (all DCs are in the LOW state), add the counters (as explained in the following) of the sender's PCL and the receiver's PCL and select the DC with the least counter.

Each SU maintains a counter for each DC, indicating the number of pairs that selected the DC for use during the next beacon interval. Once the receiver selects a DC, it sends an ATIM-ACK back to the sender, indicating the selected DC. The sender then replies with an ATIM-RES (reservation) packet to allow neighbor SUs to learn of the fact that this DC will be used by those SUs during the next beacon interval. This information is needed so that those neighbor SUs can accordingly update their NAVs.

4.2.3 Ideal Media Access Control

To develop a comparative feel, we evaluate and compare the performance of OS-MAC, R-MAC, and MC-MAC with respect to an Ideal-MAC protocol. A protocol is considered to be an Ideal-MAC when:

- It equally distributes all sessions over all DCs. That is, the total traffic load generated by all sessions is equally distributed over all DCs.
- All packets are successfully delivered at their first trial. That is, no retransmission is needed (no packet collision and no packet loss).
- There is no need for ACKs. That is, the Ideal-MAC protocol is perfect not to rely on ACKs.

Clearly, no protocol can achieve the performance obtainable under the Ideal-MAC protocol. It represents an upper bound on the average achievable performance. We will use the Ideal-MAC protocol as a baseline for our performance comparison.

4.3 Performance Metrics

We consider three performance metrics to evaluate OS-MAC from both the user's and the network's perspectives. From the user's perspective, we evaluate the SUs' per-session quality under the three network scenarios discussed above. Note that we will be concerned only with SUs' session quality. PUs always have the exclusive right to access their DC, and hence, their session quality should not be affected by the protocol being used. Therefore, to evaluate the performance of OS-MAC from the user's perspective, we evaluate two metrics.

4.3.1 Relative Session Delays $\mathcal{D}$

Under Ideal-MAC, the average session duration can be expressed as

$$\text{Ideal-MAC duration} = \frac{ZM}{NB(1-\eta_P)}.$$

We then define the delay of a session to be the time difference between its measured duration and its Ideal-MAC duration. The metric $\mathcal{D}$ of a given session is the ratio of its delay to its Ideal-MAC duration.

4.3.2 Normalized Session Goodput Shares $\mathcal{S}$

The session goodput share is defined as the fraction of the time used by a session to successfully communicate packets to that of its total measured duration. Note that we only measure the goodput; that is, we do not consider retransmission packets nor ACKs. The average session goodput share obtainable under the Ideal-MAC protocol can be expressed as

$$\text{Ideal-MAC goodput share} = \frac{N}{M}(1-\eta_P).$$

The metric $\mathcal{S}$, which evaluates the goodput of the proposed OS-MAC, is then defined as the session goodput share normalized to that obtainable under the Ideal-MAC protocol.

From the network's perspective, we consider measuring how much of the spectrum left unused by PUs can actually be exploited by SUs under each protocol. Hence, we evaluate the following metric.

4.3.3 Unused Spectrum Utilization ($\mathcal{U}$)

This metric is defined to be the ratio of bandwidth used by SUs to that of the total spectrum left unused by PUs.

Before delving into the details of the simulation results and analysis, it is important to note that when the network is lightly loaded,[7] all protocols perform well. This is due primarily to the fact that when sessions in the network are not that many, even when sessions are not perfectly balanced across the DCs, the network (that is, all DCs) can fully support all the sessions. Therefore, in this paper, we focus on networks that experience medium to high traffic loads. In the remainder of this section, we only present and analyze results when the network load is above 40 percent.

The x-axis of all plots in this section represents the traffic load generated by all SUs (η_S) normalized to the total amount of the spectrum left unused by PUs. It is called the *Secondary Traffic Load on Unused Spectrum*, denoted as η'_S. Hence, $\eta'_S = \eta_S \times \frac{1}{1-\eta_P}$. For example, when the PUs' load is $\eta_P = 60\%$ and the SUs' total load is $\eta_S = 20\%$, the x-axis point corresponding to this scenario is $\eta'_S = 50\%$.

4.4 Session Delay Analysis

In this section, we measure 1) the average session delay $\bar{\mathcal{D}}$ and 2) the coefficient of variation $\delta_{\mathcal{D}}$ of delays of all serviced sessions under all three protocols: OS-MAC, R-MAC, and MC-MAC. Although $\bar{\mathcal{D}}$ evaluates how well the protocol performs on the average, the metric $\delta_{\mathcal{D}}$ allows us to evaluate the protocol's performance in terms of fairness among multiple sessions. That is, the lower $\delta_{\mathcal{D}}$ is, the fairer the protocol becomes.

4.4.1 Average Relative Session Delay

Fig. 3 shows the relative delays averaged over all serviced sessions as a function of the SUs' traffic load η'_S under each of the three network scenarios: no PUs (Fig. 3a), PUs with $\eta_P = 30\%$ (Fig. 3b), and PUs with $\eta_P = 60\%$ (Fig. 3c).

When PUs are not present (Fig. 3a), whereas all three protocols cause similar session delays under light traffic loads, OS-MAC performs better than the other two

---

[7] Typically, and as indicated by our simulation, when $\eta <\approx 40\%$.

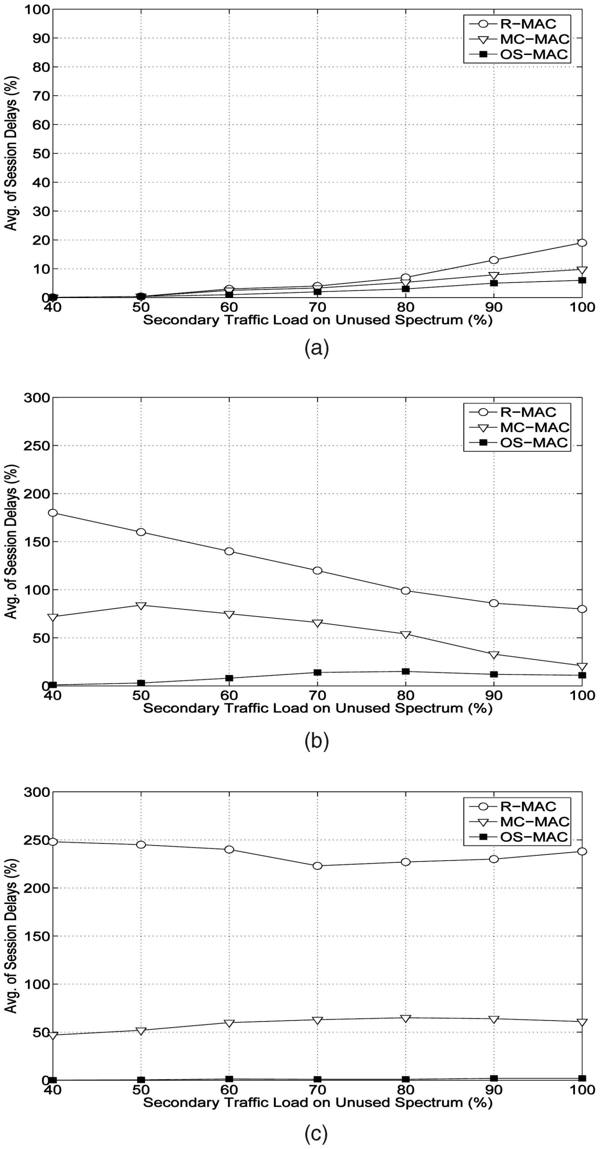


Fig. 3. Average relative session delays $\bar{\mathcal{D}}$ as a function of SUs' traffic load on unused spectrum η'_S. (a) No PUs. (b) PUs with $\eta_P = 30\%$. (c) PUs with $\eta_P = 60\%$.

protocols under medium to high loads. MC-MAC, however, still outperforms R-MAC due to its load-balancing feature. OS-MAC outperforms MC-MAC because of its adaptability. Recall that the length of beacon intervals of MC-MAC is fixed a priori. Hence, unlike OS-MAC, MC-MAC does not adapt to traffic load variations.

When PUs are present (Figs. 3b and 3c), observe that the average measured delay under R-MAC is significantly higher than that under OS-MAC. For example, Fig. 3b shows that the average duration of sessions serviced under R-MAC varies from almost three times as long ($\bar{\mathcal{D}} \approx 180$ for $\eta'_S = 40\%$) to almost twice as long ($\bar{\mathcal{D}} \approx 80\%$ for $\eta'_S = 95\%$) as the average duration obtained under Ideal-MAC. This delay is even longer when PUs incur higher loads, as in the case shown in Fig. 3c ($\eta_P = 60\%$), where the session duration could be almost three and a half times as long as the Ideal-MAC duration. Compared to MC-MAC, OS-MAC also achieves better performance in terms of session delays. Note that sessions under MC-MAC can be delayed for almost one and a half times as long as in the case of Ideal-MAC ($\bar{\mathcal{D}} \approx 50\%$). On the other hand, the delays measured under OS-MAC are significantly small (always less than 5 percent) regardless of the traffic load. It is also worth noting that these results show that the proposed protocol not only outperforms R-MAC and MC-MAC but also performs almost as well as Ideal-MAC.

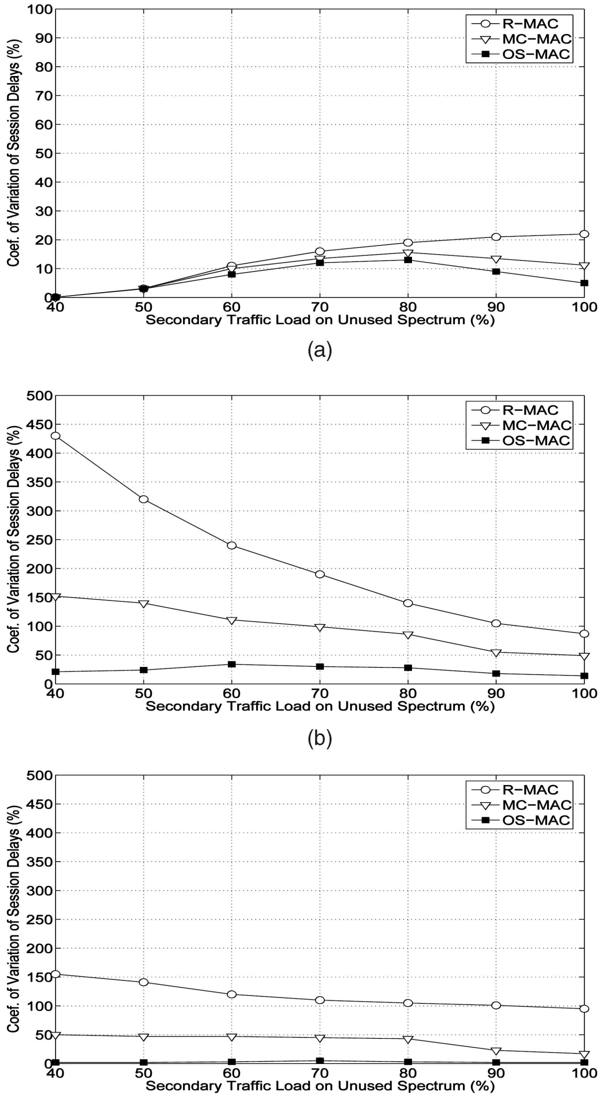
The performance difference between OS-MAC and the other two protocols is due to its two distinct features: adaptation and selection. First, unlike MC-MAC, the parameter `SelWin` of OS-MAC dynamically adapts to channel conditions and load. The higher the variability of traffic load across DCs is, the smaller `SelWin` becomes, and vice versa. Hence, when DCs' loads are highly variable, OS-MAC shortens its period so that SUs can seek better DCs often enough to exploit unused spectrum opportunities. On the other hand, when DCs experience less load variability (all DCs have similar loads), the length of the period accordingly augments to avoid unnecessary switching. Second, the metric (access-time share or throughput share) through which OS-MAC assesses the condition of a given DC accounts for the load incurred by PUs, whereas MC-MAC decides purely based on the number of SUs that currently use a given DC.

4.4.2 Coefficient of Variation of $\mathcal{D}$: A Fairness Index

Fig. 4 shows the coefficient of variation of all measured average session delays as a function of SUs' traffic load (η'_S), again under each of the three network scenarios: no PUs (Fig. 4a), PUs with $\eta_P = 30\%$ (Fig. 4b), and PUs with $\eta_P = 60\%$ (Fig. 4c). Recall that the coefficient of variation $\delta_{\mathcal{D}}$ of session delays is a way of measuring and evaluating the protocols' fairness with respect to the time (duration) for sessions to complete. That is, higher values of $\delta_{\mathcal{D}}$ imply that the corresponding protocol does not fairly service all sessions.

There are two observations to make on the fairness of a protocol. First, sessions supported under OS-MAC not only take, on the average, no longer than those obtainable under R-MAC or MC-MAC, but also are equally treated by finishing each within a time that is proportional to its size. Moreover, this fair treatment by OS-MAC is always assured, regardless of the PUs' load, whereas it is not assured under R-MAC nor MC-MAC. In fact, as shown in Fig. 4c, the variation of session delays under R-MAC ranges from almost $\delta_{\mathcal{D}} = 150\%$ to almost $\delta_{\mathcal{D}} = 100\%$ when η'_S varies from 40 percent to 100 percent. This means that under R-MAC, some sessions could be delayed more than twice as long as other sessions. In certain situations, as in the case shown in Fig. 4b, when $\eta'_S \simeq 50\%$, some sessions could be delayed almost four times longer than other sessions. The variation of session delays under MC-MAC is also higher than that under OS-MAC. It could be as high as $\delta_{\mathcal{D}} = 150\%$ when $\eta_P = 30\%$ (Fig. 4b) and as high as $\delta_{\mathcal{D}} = 50\%$ when $\eta_P = 60\%$ (Fig. 4c). In summary, in terms of delay variation among different sessions, the proposed protocol is not only fairer than R-MAC and MC-MAC but is also fair since the obtained coefficients of variation are very small.

Second, the variation of delays of different sessions is sensitive to the total traffic load, independent of the

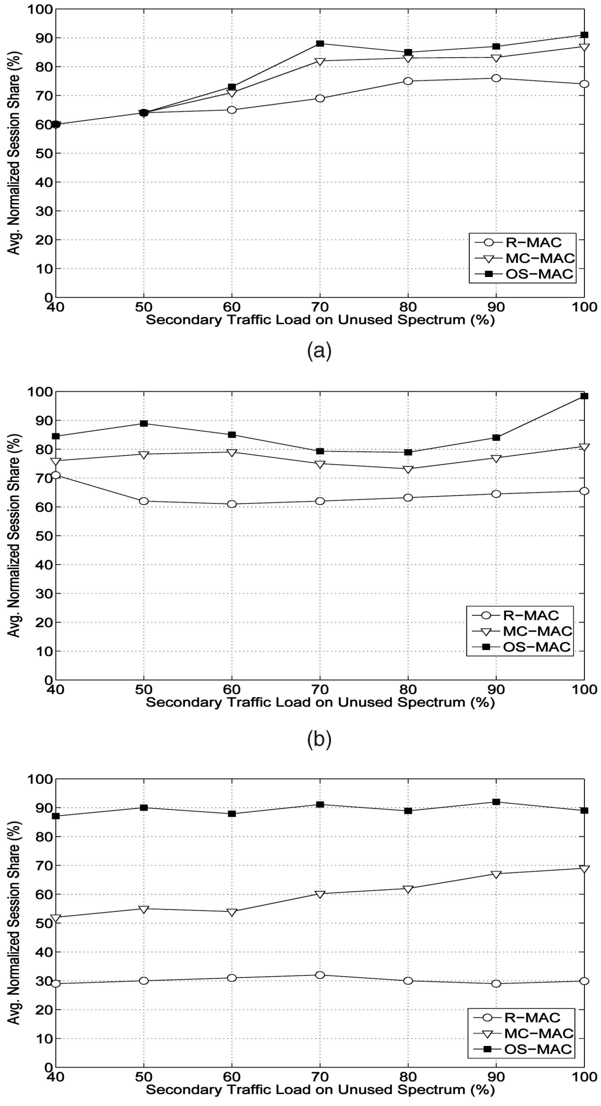


(a)

(b)

(c)

Fig. 4. Coefficient of variation of relative session delays δ_D as a function of SUs' traffic load on unused spectrum η'_S. (a) No PUs. (b) PUs with $\eta_P = 30\%$. (c) PUs with $\eta_P = 60\%$.



(a)

(b)

(c)

Fig. 5. Average normalized session goodput shares $\bar{S}$ as a function of SUs' traffic load on unused spectrum η'_S. (a) No PUs. (b) PUs with $\eta_P = 30\%$. (c) PUs with $\eta_P = 60\%$.

protocol being used, and behaves as follows: When the total network load η is high or low, the variation is low. This is because, regardless of how well the protocol balances the sessions across the DCs, at high loads, all sessions end up getting delayed, making the delay variation small. If the network load is low, the delay variation is also small, but this time, the network is able to support all sessions with almost no or little delay.

Recall that the total network load η is equal to $\eta_P + \eta_S = \eta_P + \eta'_S(1 - \eta_P)$. For example, an SU traffic load on the unused spectrum of, say, $\eta'_S = 40\%$ corresponds to a total network load of $\eta = 65\%$ (Fig. 4b) and $\eta = 80\%$ (Fig. 4c) if the PUs' network load is $\eta_P = 30\%$ and $\eta_P = 60\%$, respectively. This explains why the $\eta'_S = 40\%$ in Fig. 4b (which corresponds to a medium total network load of $\eta = 65\%$) results in a higher delay variation than the $\eta'_S = 40\%$ (which corresponds to a high total network load of $\eta = 80\%$) in Fig. 4c. Clearly, the $\eta'_S = 40\%$ in Fig. 4a still corresponds to a low total network load of $\eta = 40\%$, since no PUs are present. This also results in a low delay variation.

4.5 Goodput Analysis

We now evaluate the performance in terms of sessions' achievable goodputs. We measure the average $\bar{S}$ of normalized (with respect to Ideal-MAC) goodput shares of all serviced sessions in the network. Although it suffices to evaluate the performance from either a delay or a throughput perspective (since, in theory, both are equivalent), we decided to present the throughput results to confirm our analysis and performance.

Fig. 5 shows the measured normalized goodput share averaged over all serviced sessions as a function of the SUs' network traffic load η'_S under each of the three network scenarios: no PUs (Fig. 5a), PUs with $\eta_P = 30\%$ (Fig. 5b), and PUs with $\eta_P = 60\%$ (Fig. 5c).

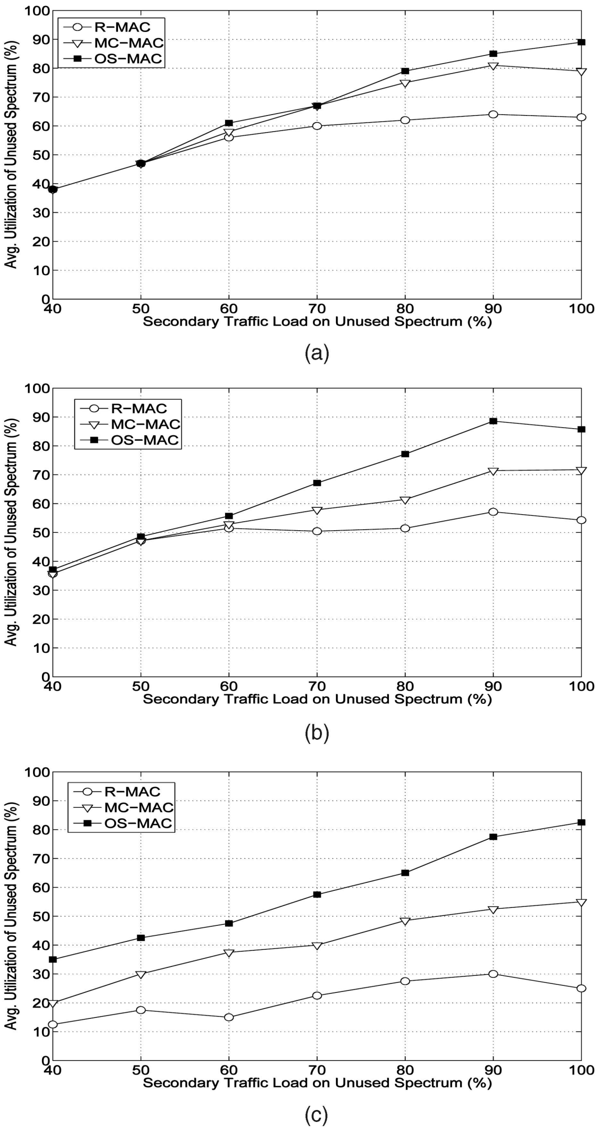
First, note that each session serviced under the proposed protocol achieves, on the average, a goodput share $\bar{S}$ of more than 85 percent of that achievable under Ideal-MAC. In addition, observe that, the higher the network load due to PUs is, the closer the achieved share under the proposed protocol is to that achievable under Ideal-MAC. This demonstrates that the proposed protocol performs extremely well, given that Ideal-MAC does not account for the bandwidth used neither for packet retransmission nor for sending ACKs. Hence, most of the difference in the achievable goodputs is actually consumed by ACKs and possible packet retransmissions due to collisions. Another point that requires attention is that the achievable goodput shares under OS-MAC are not sensitive to the PUs' network load. This is an important feature of OS-MAC. Both R-MAC and MC-MAC, on the other hand, provide sessions with lesser shares than what they would otherwise achieve under Ideal-MAC, especially for networks with medium or high traffic loads. For example, although the normalized average goodput share is $\approx 60\%$ for R-MAC and 75 percent for MC-MAC when $\eta_P = 30\%$, it is only $\approx 30\%$ for R-MAC and 60 percent for MC-MAC when $\eta_P = 60\%$. Unlike OS-MAC, both R-MAC and MC-MAC are sensitive to network traffic loads.

4.6 Unused Spectrum Analysis

To evaluate the performance of OS-MAC from the network's perspective, we measure the percentage of the bandwidth/spectrum that is actually used by SUs to that of the total spectrum left unused by PUs. Fig. 6 shows this percentage as a function of the SUs' network traffic load η'_S under each of the three network scenarios: no PUs (Fig. 6a), PUs with $\eta_P = 30\%$ (Fig. 6b), and PUs with $\eta_P = 60\%$ (Fig. 6c).

First, note that, independent of PUs' network traffic load, SUs under OS-MAC utilize the spectrum left unused by PUs to its fullest extent. For example, SUs' network traffic load of 40 percent ($\eta'_S = 40\%$) also yields to about 35 percent to 40 percent of utilization of the total spectrum left unused by PUs, whereas SUs' network traffic load of 90 percent ($\eta'_S = 90\%$) also yields about 85 percent of utilization of the total spectrum left unused by PUs. This is true, regardless of the PUs' network traffic load (see Fig. 6a for PUs' traffic load $\eta_P = 0$, Fig. 6b for PUs' traffic load $\eta_P = 30\%$, and Fig. 6c for PUs' traffic load $\eta_P = 60\%$). In addition, observe that OS-MAC is not sensitive to PUs' network loads; that is, the spectrum left unused by PUs is fully exploited by SUs under each of the three network scenarios: $\eta_P = 0\%$ as in Fig. 6a, $\eta_P = 30\%$ as in Fig. 6b, and $\eta_P = 60\%$ as in Fig. 6c. Hence, OS-MAC performs well not only from the user's perspective but also from the network's perspective.

When the network load is medium to heavy (that is, the SUs' traffic load is greater than 80 percent and the PUs' traffic load is greater than 60 percent, as shown in Fig. 6c), note that R-MAC and MC-MAC respectively result in an average utilization of the unused spectrum of only about 25 percent to 30 percent and 55 percent to 60 percent, whereas OS-MAC always results in an average spectrum utilization of more than 85 percent. The performance difference between OS-MAC and MC-MAC in terms of spectrum utilization is due to 1) the adaptation of OS-MAC's parameters to channel conditions and loads, 2) the PU-aware channel assessment metric of OS-MAC, and 3) the delegation mechanism of OS-MAC that, unlike the case of MC-MAC, avoids having all SUs periodically switch to the CC for channel setups and selection. Having all SUs switch to the CC every beacon interval, as in MC-MAC, results in bandwidth wastage since all DCs will not be used during the entire ATIM window period. In our protocol, only one delegate of each DC switches to the CC, whereas all other SUs remain in and continue using their DCs.



(a)

(b)

(c)

Fig. 6. Average utilization of unused spectrum $\bar{U}$ as a function of SUs' traffic load on unused spectrum η'_S. (a) No PUs. (b) PUs with $\eta_P = 30\%$. (c) PUs with $\eta_P = 60\%$.

Based on the simulation results, we can make two claims. First, OS-MAC is shown to be more effective than R-MAC and MC-MAC not only from the network's perspective but also from the user's perspective. Second, it achieves performances that are comparable to those obtainable under Ideal-MAC.

4.7 Extensions to OS-MAC

The *access-time share* metric used by OS-MAC to characterize a DC's conditions is defined as the *ratio* of the amount of time during which the SU possesses the channel during the Select Phase to that of the total duration of the Select Phase. This ratio is periodically updated every OSP. Note that when all DCs support the same data rate, the access-time share is exactly equivalent to the *obtainable throughput share*, which can simply be calculated as the access-time share times the bandwidth of the DC. Hence, the metric used by OS-MAC to assess channel conditions ensures that each user receives an equal share of the available throughput.

We now show how OS-MAC can easily be extended to support the case when different DCs may be allocated different bandwidths or experience different channel conditions, thereby enabling them to support different data rates. In such a case, we define the *obtainable throughput share* as the total number of bits successfully sent during the Select Phase divided by the length of the Select Phase, and we use it for assessing the condition of a channel. Note, however, that this incurs very little modification to OS-MAC. Each user will then have to count the number of successfully delivered packets in lieu of the amount of time during which it had control over the DC.

5 PRACTICALITY VERSUS EFFICIENCY

5.1 Is a Common Control Channel Necessary?

The designation of a portion of resource as a "common good" may appear unattractive to selfish individuals. Are we not fortunate that this does not always hold? In some cases, all individuals will be better off with the "common good" than if each had pursued only his selfish interest. Public parks and highways are two illustrative examples of "common goods" (the land is the resource here), where all individuals would be worse off without them. Imagine what would happen if there were no highways, but each individual had a piece of land. With some reflection, one can observe that the inefficiency of the spectrum resource is pretty much due to the lack of efficient access methods, which is, in turn, due to the spectrum's current selfish command-and-control regulations. Dedicating a piece of the spectrum as a common means for collaborative tasks is, indeed, an absolute necessity to achieve spectrum access efficiency. To a considerably large extent, each having a spectrum band without a common channel is very much like each having a piece of land without a highway.

Moreover, previous studies [15], [16], [17] show and argue that the dedication of a common channel leads to high overall spectrum efficiency. In [15] and [16], the authors experimentally show that a common spectrum coordination channel (CSCC) actually significantly improves the overall efficiency of the spectrum. A second case where a common channel is shown to be very beneficial is the European DRiVE Project [17], in which a dedicated common channel, referred to as a logical common coordination channel (LCCC), is used as a means for spectrum users to coordinate for better dynamic spectrum allocation. From its efficient usage standpoint, the spectrum is far better off with a dedicated common channel than without it. The design of OS-MAC is based on this principle.

5.2 Will Cooperation Prevail?

One subtle question in spectrum agility that has not yet fully been answered is how PUs can be protected from the SUs' interference if SUs are allowed to opportunistically access and use their spectrum. To make the matter even worse, the issue is not so much of how interference protection can be assured but how we can do so while maximizing spectrum utilization. Let us think of the question as a two-step challenge: first, get it to work, and then, make it efficient.

It is worth noting that the above interference problem cannot efficiently be solved unless SUs are capable of detecting the presence or the return of PUs in any spectrum band that they use or may use. As mentioned in Section 2, this work assumes that the underlying physical radio is capable of detecting the return or the presence of PUs. Hence, the physical layer is assumed to inform its MAC layer of its detection of PUs. Under this assumption, we now look at what OS-MAC can do to tackle the above two-step challenge.

If we relax the efficiency requirement, it should be apparent that interference can be avoided by just empowering SUs with the capability of vacating the licensed spectrum upon the detection of the return or the presence of PUs. One simple approach is then to have SUs switch to the common channel and start over after finding new spectrum opportunities. Another even simpler approach is to have SUs cease communication upon the detection of PUs and stay tuned to the same spectrum band until it becomes vacant again. Clearly, these two approaches are not efficient from the spectrum utilization's standpoint. If spectrum efficiency is our ultimate goal and hence presents a constraint to the problem, then the MAC layer must also provide SUs with a *low-cost recovery* mechanism. The recovery mechanism should allow SUs to *quickly* find and switch to different spectrum bands upon the detection of the return of PUs.

There are two approaches that OS-MAC can use to efficiently solve the interference-avoidance problem: *noncooperative* and *cooperative*. In the noncooperative approach, OS-MAC assumes that PUs do not cooperate with SUs for better spectrum utilization. For example, PUs do not alert SUs of their return nor permit spectrum sharing with SUs, even for an amount of time that allows SUs to inform each other of and switch to a new opportunistic band. In this case, SUs must cease using the licensed spectrum band immediately upon the detection of PUs. In OS-MAC, all SUs must suspend their sessions, switch to CC, and wait until the next Update Phase to select a new DC. Although the reason behind the immediate vacancy is to preserve the PUs' quality of service, this may cause the overall spectrum inefficiency, which unfortunately conflicts with the main objective. We envision that this approach is likely to be more applicable and attractive from the implementation/practicality point of view, at least in a short-term strategy, since it does not require explicit involvement of licensees.

The cooperative approach, on the other hand, consists of having spectrum licensees collaborate with SUs to achieve efficient spectrum use. For example, if SUs are allowed to continue using the spectrum after the detection of PUs for a short duration of time before vacating the channel, they may be able to inform each other of other potential spectrum opportunities and, hence, seize and switch to one of them without going through the common channel. In fact, one can think of several ways to improve spectrum use, each of which depends not only on licensees' willingness and incentives to collaborate but also on their spectrum access methods. For example, if PUs use the CDMA technology as their access method, we argue that it would be more beneficial in terms of the overall spectrum efficiency if a few spreading codes are reserved for SUs to use in such an emergency case. On the other hand, if PUs rely on the OFDMA technology to access their spectrum, one can reserve one narrow-band channel for SUs to use in case of the return of PUs. In essence, we think that spectrum use can be far more effective if a small portion of the bandwidth of each spectrum band (whether time, frequency, or code) is reserved for emergency use. Obviously, this approach requires that spectrum policies and market regulations evolve toward more flexible models than current ones. Spectrum policy makers are then required to implement such flexible strategies not only in newly allocated bands but also and gradually in the already-occupied spectrum. It requires 1) intelligent economical/pricing strategies to encourage licensees to move toward spectrum openness and 2) innovative transitional mechanisms that can be employed to improve spectrum efficiency without degrading the quality of existing services.

Although less efficient, we envision that the noncooperative approach is a short-term solution for the opportunistic access of spectrum. The cooperative approach will prevail in a longer term.

6 RELATED WORK

There have been numerous studies on the classical multiple channel access [14], [18], [19], [20], [21]. In general, most of the reported protocols set aside a channel (CC) for traffic control and use the others for data communications (DCs). DCA-MAC, as proposed in [19], assumes that each wireless device is equipped with two half-duplex transceivers: One is always tuned on the CC and the other is tuned on a DC. DCA-MAC operates on a packet-by-packet basis; that is, prior to transmitting each packet, the source-destination pair must switch to the CC to negotiate the new DC, on which the next packet will take place. To some extent, DCA-MAC is very much like the IEEE 802.11, except that the RTS/CTS handshaking mechanism indicates not only who is using the medium during next packet but also who is using each DC. Another multiple-channel MAC, called MC-MAC, is proposed in [14]. Like OS-MAC, MC-MAC also uses one half-duplex transceiver. In MC-MAC, all devices must periodically tune on the CC for an interval of time, called ATIM, during which source-destination pairs negotiate and select their new DC. The period, which is a design parameter, is chosen and fixed at the time of initialization. Most of these reported multiple-channel access protocols cannot be used in the context of spectrum agility for several reasons. First, they are not designed for opportunistically accessing licensed spectrum. Second, they are mostly designed for conventional one-to-one packet communication, and hence, they do not support the notion of a group of users involved with a session. Finally, they are static in the sense that their parameters are to be fixed a priori, and hence, they do not adapt to current traffic loads for better spectrum utilization in real time.

The design of dynamic and agile techniques for spectrum sharing and allocation is more recent and is still in its infancy [22], [23], [24], [25], [26], [27], [28]. Generally, these reported techniques can be classified into two categories: centralized [22], [23] and distributed [24], [25], [26], [27], [28]. In [22], a centralized protocol, called the Dynamic Spectrum Access Protocol (DSAP), for managing and coordinating spectrum access is proposed. The DSAP is basically a way of providing and managing the dynamic allocation of spectrum bands to users while avoiding congestion and minimizing interference. Zekavat and Li [23] propose a new cognitive radio-based architecture for dynamic channel allocation. The basic idea is that, instead of having users always subscribe to and receive service from one service provider, they can dynamically and adaptively change their service provider based on quality/cost metrics such as channel availability, congestion, and cost. These approaches are new and interesting concepts to address the inefficient and "static" way of current spectrum allocation policies.

There have also been distributed approaches ranging from dynamic allocation of spectrum [24], [25] to its adaptive sharing among multiple users [26], [27], [28]. Cao and Zheng [24] propose a dynamic bargaining approach for spectrum allocation across mobile users. The approach extends existing graph-coloring-based spectrum assignment schemes to account for mobility. Their approach reduces computation and communication overheads by taking prior-to-move allocation information into account in determining the new assignments. Along the same line, Zhao et al. [25] present a dynamic channel coordination scheme, where users organize themselves into groups with similar communication interests. Although members of each group subscribe and use one channel to communicate with each other, boundary members are allowed to subscribe to multiple channels to maintain connectivity across multiple groups. In [26], the channel allocation problem is modeled to be the outcome of a game in which the players are the users, their actions are the choices of transmitting channels, and their preferences are reflected through the quality of the chosen channels. They also define two different objective functions for the spectrum sharing games, respectively capturing the utility of selfish and cooperative users. Based on these game-theoretic approaches, cooperation-based spectrum sharing methods are shown to achieve better spectrum access performance than noncooperative sharing ones. This, however, comes at the expense of increased overhead due to required information exchange. Sankaranarayanan et al. [27] propose AS-MAC, a multiband access MAC protocol that enables communication between pairs of nodes. Basically, AS-MAC empowers nodes to first agree upon a DC through a handshake that involves the exchange of three control messages—an RTS, a CTS, and a Reservation (RES) message—and then switch to it for communication. This handshake is similar to that of the IEEE 802.11, except that instead of agreeing on which time slots to reserve, pairs of nodes use the handshake to agree and then reserve the DC to communicate on. In [28], a MAC protocol, called DOSS, is proposed for spectrum-agile networks. Like DCA-MAC, DOSS functions on a packet-by-packet basis; a new channel is negotiated for each packet. Under DOSS, the spectrum is divided into one CC and many pairs of (data or busy-tone) bands; that is, for each data band, there is a busy tone band mapped to it. Although the CC is used for DC negotiation, busy-tone bands are used by receivers to prevent nearby transmitters from interfering with them. DOSS then prevents this interference by requiring that receivers continuously send busy tones on the corresponding busy tone bands during the whole course of their receptions. DOSS has three major disadvantages. First, although busy tones solve the hidden-terminal problem with lesser traffic overhead than the traditional IEEE 802.11 RTS/CTS handshaking mechanism, the bandwidth that they use may be significant, thereby resulting in spectrum inefficiency. Second, DOSS requires that each device has at least two transceivers: one for sending busy tones and the other for data communication. Finally, power consumption may now present a major concern due to the extra amount of energy needed for transmissions of busy tones.

7 CONCLUSION

In this paper, we proposed the OS-MAC protocol for cognitive wireless networks. OS-MAC dedicates one channel as a common CC, where interchannel control traffic takes place. In OS-MAC, devices are only required to be equipped with a half-duplex transceiver. OS-MAC empowers SDR-based wireless devices with the capabilities of 1) adaptively and dynamically seeking and exploiting opportunities in both licensed and unlicensed spectra and along both the time and the frequency dimensions, 2) accessing and sharing spectrum among different unlicensed and licensed users, and 3) coordinating with other unlicensed users for better spectrum utilization.

OS-MAC has several distinct features. First, it significantly improves the spectrum access efficiency by balancing the traffic load over all spectrum bands. Second, it fairly treats all users by assuring them to receive an equal access-time share or throughput share. Third, it incurs no or little control overhead. Finally, it dynamically adapts to the network traffic load to achieve higher performance while minimizing the control overhead.

The performance of OS-MAC is evaluated using an ns2-based simulation. We showed that OS-MAC is far more effective than current protocols from both the network's and the user's perspectives. We also showed that OS-MAC achieves performances that are very close to those achievable under the Ideal-MAC protocol.

APPENDIX

INTRODUCTION OF IEEE 802.11

The IEEE 802.11 MAC [13] protocol supports two types of traffic: asynchronous and synchronous. The protocol allows simultaneous existence of both types by partitioning transmission time units, called superframes, into a *contention-free period* (CFP) and a *contention period* (CP). The point coordination function (PCF) is an access method provided by the IEEE 802.11 Standard to support the synchronous traffic during CFPs, whereas the distributed coordination function (DCF) is an access method that the Standard provides to support the asynchronous traffic during CPs. The DCF method is based on the CSMA paradigm and is originally designed to solve and tackle certain problems, namely, the hidden and exposed terminal problems, which are introduced by the wireless nature of the ad hoc networks. The IEEE 802.11 DCF Standard Specifications then included the collision-avoidance feature by means of the RTS and CTS handshake mechanism to solve these problems.

According to the DCF specifications, prior to transmitting a packet, a user must first sense the medium to be idle for a minimum duration called the DIFS period. Then, to reduce collision, the user must wait for an additional random backoff period calculated as $b \times \tau$, where b is a number, called *backoff counter*, selected from a uniform distribution in the interval $[0, W_0 - 1]$ and τ is the length of the time slot period. The parameter W_0 is a fixed number referred to as the initial *contention window* size. While waiting, the user decrements its counter by 1 every idle time slot. Every time the medium becomes busy, the user must freeze its backoff counter. Once the counter is frozen, the user resumes decrementing the counter by 1 every idle time slot after sensing that the medium is again idle for a DIFS period. When the counter reaches 0, the user proceeds transmission. In the case of unsuccessful transmission, the user keeps retransmitting the packet until it either succeeds or reaches a threshold number of attempts. At the $i\mathrm{th}$ retransmission attempt, the contention window size W must equal $W_i = max\{f^i \times W_0, W_m\}$, where f is a persistent factor (typically $f = 2$) and W_m is the maximum allowed size of the contention window. Upon a successful transmission, the contention window is rest to its initial size. When the receiving user receives a nonerroneous packet, it only needs to wait for a short interframe space (SIFS) period—shorter than the DIFS period—before acknowledging the sender.

REFERENCES

[1] XG Working Group, *The XG Vision,* RFC v2.0, 2007.

[2] D. Schaefer, "Wide Area Adaptive Spectrum Applications," *Proc. IEEE Military Comm. Conf. (MILCOM '01),* Oct. 2001.

[3] W. Horne, P. Weed, and D. Schaefer, "Adaptive Spectrum Radio: A Feasibility Platform on the Path to Dynamic Spectrum Access," *Proc. 10th Ann. Int'l Symp. Advanced Radio Technologies (ISART '03),* Mar. 2003.

[4] *IEEE 802.22,* www.ieee802.org/22, 2007.

[5] C. Cordeiro, K. Challapali, and S. Shankar, "IEEE 802.22: The First Worldwide Wireless Standard Based on Cognitive Radios," *Proc. First IEEE Int'l Symp. New Frontiers in Dynamic Spectrum Access Networks (DySPAN '05),* Nov. 2005.

[6] *Report ET Docet 02-135,* Spectrum Policy Task Force (SPTF), FCC, Nov. 2002.

[7] *Report of the Spectrum Efficiency WG,* Spectrum Policy Task Force (SPTF), FCC, Nov. 2002.

[8] M. McHenry, *Reports on Spectrum Occupancy Measurements,* Shared Spectrum Company, www.sharedspectrum.com/?section=nsf_summary, 2007

[9] D. Cabric, S. Mubaraq, and R.W. Brodersen, "Implementation Issues in Spectrum Sensing for Cognitive Radios," *Conf. Record 38th Asilomar Conf. Signals, Systems and Computers,* pp. 772-776, Nov. 2004.

[10] B. Wild and K. Ramchandran, "Detecting Primary Receivers for Cognitive Radio Applications," *Proc. First IEEE Int'l Symp. New Frontiers in Dynamic Spectrum Access Networks (DySPAN '05),* pp. 124-130, Nov. 2005.

[11] G. Ganesan and Y.G. Li, "Cooperative Spectrum Sensing for Opportunistic Access in Fading Environments," *Proc. First IEEE Int'l Symp. New Frontiers in Dynamic Spectrum Access Networks (DySPAN '05),* pp. 137-143, Nov. 2005.

[12] S. Shankar, C. Cordeiro, and K. Challapali, "Spectrum Agile Radios: Utilization and Sensing Architectures," *Proc. First IEEE Int'l Symp. New Frontiers in Dynamic Spectrum Access Networks (DySPAN '05),* pp. 160-169, Nov. 2005.

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

[14] J. So and N. Vaidya, "Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using a Single Transceiver," *Proc. ACM MobiHoc,* pp. 222-233, 2004.

[15] X. Jing and D. Raychaudhuri, "Spectrum Co-Existence of IEEE 802.11b and 802.16a Networks Using the CSCC Etiquette Protocol," *Proc. First IEEE Int'l Symp. New Frontiers in Dynamic Spectrum Access Networks (DySPAN '05),* pp. 243-249, Nov. 2005.

[16] D. Raychaudhuri and X. Jing, "A Spectrum Etiquette Protocol for Efficient Coordination of Radio Devices in Unlicensed Bands," *Proc. 14th IEEE Personal Indoor and Mobile Radio Conf.*, 2003.

[17] L. Xu, R. Tonjes, T. Paila, W. Hansmann, M. Frank, and M. Albrecht, "DRiVE-ing to the Internet: Dynamic Radio for IP Services in Vehicular Environments," *Proc. 25th Ann. IEEE Conf. Local Computer Networks (LCN '00)*, pp. 281-289, Nov. 2000.

[18] M. Kodialam and T. Nandagopal, "Characterizing the Capacity Region in Multi-Radio Multi-Channel Wireless Mesh Networks," *Proc. ACM MobiCom*, pp. 73-87, 2005.

[19] S. Wu, C. Lin, Y. Tseng, and J. Sheu, "A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Multi-Hop Mobile Ad Hoc Networks," *Proc. Int'l Symp. Parallel Architectures, Algorithms and Networks (ISPAN '00)*, pp. 232-237, 2000.

[20] A. Nasipuri and S.R. Das, "Multichannel CSMA with Signal Power-Based Channel Selection for Multihop Wireless Networks," *Proc. 52nd IEEE Vehicular Technology Conf. (VTC '00-Fall)*, 2000.

[21] P. Kyasanur and N.H. Vaidya, "Capacity of Multi-Channel Wireless Networks: Impact of Number of Channels and Interfaces," *Proc. ACM MobiCom*, pp. 43-57, 2005.

[22] V. Brik, E. Rozner, S. Banarjee, and P. Bahl, "DSAP: A Protocol for Coordinated Spectrum Access," *Proc. First IEEE Int'l Symp. New Frontiers in Dynamic Spectrum Access Networks (DySPAN '05)*, pp. 611-614, 2005.

[23] S.A. Zekavat and X. Li, "User-Central Wireless System: Ultimate Dynamic Channel Allocation," *Proc. First IEEE Int'l Symp. New Frontiers in Dynamic Spectrum Access Networks (DySPAN '05)*, pp. 82-87, 2005.

[24] L. Cao and H. Zheng, "Distributed Spectrum Allocation via Local Bargaining," *Proc. Second Ann. IEEE Conf. Sensor and Ad Hoc Comm. and Networks (SECON '05)*, pp. 475-486, 2005.

[25] J. Zhao, H. Zheng, and G.-H. Yang, "Distributed Coordination in Dynamic Spectrum Allocation Networks," *Proc. First IEEE Int'l Symp. New Frontiers in Dynamic Spectrum Access Networks (DySPAN '05)*, pp. 259-268, 2005.

[26] N. Nie and C. Comaniciu, "Adaptive Channel Allocation Spectrum Etiquette for Cognitive Radio Networks," *Proc. First IEEE Int'l Symp. New Frontiers in Dynamic Spectrum Access Networks (DySPAN '05)*, pp. 269-278, 2005.

[27] S. Sankaranarayanan, P. Papadimitratos, A. Mishra, and S. Hershey, "A Bandwidth Sharing Approach to Improve Licensed Spectrum Utilization," *Proc. First IEEE Int'l Symp. New Frontiers in Dynamic Spectrum Access Networks (DySPAN '05)*, pp. 279-288, 2005.

[28] L. Ma, X. Han, and C.C. Shen, "Dynamic Open Spectrum Sharing MAC Protocol for Wireless Ad Hoc Networks," *Proc. First IEEE Int'l Symp. New Frontiers in Dynamic Spectrum Access Networks (DySPAN '05)*, pp. 203-213, 2005.

Bechir Hamdaoui received the Diploma of Graduate Engineering from the National School of Engineers at Tunis (BAC+6+DEA, ENIT), Tunisia, in 1997 and the MS degree in electrical and computer engineering, the MS degree in computer sciences, and the PhD degree in computer engineering from the University of Wisconsin, Madison, in 2002, 2004, and 2005, respectively. From 1998 to 1999, he was a quality control and planning engineer on a power generation plant project under the supervision of PIRECO/FiatAvio. During the summer of 2004, he was an intern with Telcordia Technologies. In September 2005, he joined the Real-Time Computing Research Laboratory (RTCL), University of Michigan, Ann Arbor, as a postdoctoral researcher. Since September 2007, he has been with the School of Electrical Engineering and Computer Science, Oregon State University, as an assistant professor. His research interest is in wireless networking, in particular, developing and designing protocols and methods that provide next-generation wireless networks with the capabilities of energy-aware routing, admission control and QoS, spectrum agile access and sharing, and efficient usage of network resources. He is a member of the IEEE and the IEEE Computer Society.

Kang G. Shin received the BS degree in electronics engineering from Seoul National University, Seoul, in 1970, and the MS and PhD degrees in electrical engineering from Cornell University, Ithaca, New York, in 1976 and 1978, respectively. From 1978 to 1982, he was with the faculty of Rensselaer Polytechnic Institute, Troy, New York. He has held visiting positions at the US Airforce Flight Dynamics Laboratory, the AT&T Bell Laboratories, the Computer Science Division, Department of Electrical Engineering and Computer Science, University of California, Berkeley, the International Computer Science Institute, Berkeley, the IBM T.J. Watson Research Center, the Software Engineering Institute, Carnegie Mellon University, and the Hewlett-Packard (HP) Research Laboratories. For three years beginning in January 1991, he was the chair of the Computer Science and Engineering Division, Department of Electrical Engineering and Computer Science, University of Michigan. He is currently the Kevin and Nancy O'Connor Professor of Computer Science and the founding director of the Real-Time Computing Laboratory, Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor. He was the general chair of the Third ACM/Usenix International Conference on Mobile Systems, Applications, and Services (MobiSys 2005), the Sixth IEEE Real-Time Technology and Applications Symposium (RTAS 2000), and the Eighth IEEE Real-Time Systems Symposium (RTSS 1987). He was the program committee chair of RTSS 1986, was a program committee cochair of the 1992 International Conference on Parallel Processing, and served on numerous technical program committees. From 1991 to 1993, he was the chair of the IEEE Technical Committee on Real-Time Systems. In August 1987, he was a guest editor of the *IEEE Transactions on Computers* (special issue on real-time systems). He was an editor of the *IEEE Transactions on Parallel and Distributed Computing* and an area editor of the *International Journal of Time-Critical Computing Systems*, *Computer Networks*, and *ACM Transactions on Embedded Systems*. His current research interests include QoS-sensitive networking and computing, embedded real-time OS, middleware, and applications, with an emphasis on timeliness and dependability. He has supervised the completion of 57 PhD dissertations and is the author/coauthor of more than 650 technical papers (235 of which are in archival journals) and numerous book chapters in distributed real-time computing and control, computer networking, fault-tolerant computing, and intelligent manufacturing. He is a coauthor (jointly with C.M. Krishna) of *Real-Time Systems* (McGraw Hill, 1997). He is also a coauthor (with his students) of papers that received the Best Student Paper Awards in RTAS 1996 and the 2000 Usenix Annual Technical Conference. He is a fellow of the IEEE, the ACM, and the IEEE Computer Society, is a member of the Korean Academy of Engineering, and was a distinguished visitor of the IEEE Computer Society. He is a recipient of several Best Paper Awards, including the William R. Bennett Prize Paper Award from the IEEE Communications Society in 2003, the Best Paper Award from the 11th International Workshop on Quality of Service (IWQoS 2003), and the Outstanding Paper Award from the IEEE Transactions on Automatic Control in 1987. He also received several institutional awards, including the Research Excellence Award in 1989, the Outstanding Achievement Award in 1999, the Service Excellence Award in 2000, the Distinguished Faculty Achievement Award in 2001, the Distinguished Alumni Award from the College of Engineering, Seoul National University, in 2002, the IEEE RTC Technical Achievement Award in 2003, the Stephen Attwood Award (the highest honor at Michigan Engineering) from the University of Michigan in 2004, and the Ho-Am Prize in Engineering (the highest honor bestowed to engineers of Korean origin) in 2006.