Extended Abstract: Towards Context-Aware Wireless Spectrum Agility

Ashwini Kumar
Real-Time Computing Laboratory
EECS Dept., The University of Michigan
Ann Arbor, MI 48109-2121, USA
ashwinik@eecs.umich.edu

Kang G. Shin
Real-Time Computing Laboratory
EECS Dept., The University of Michigan
Ann Arbor, MI 48109-2121, USA
kgshin@eecs.umich.edu

ABSTRACT

Spectrum agility (SA) is a novel way of improving spectrum utilization efficiency and making greater bandwidth available to network applications. Despite its advantages, the fundamental operations (e.g., periodic spectrum sensing) involved in realizing SA can also adversely affect application performance. We propose *Context-Aware Spectrum Agility* (CASA), which uses application hints together with the current channel condition to adapt key SA parameters. Our preliminary evaluation shows that CASA improves throughput by 35%, and is also more effective in meeting application demands than conventional SA.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*

General Terms

Algorithms, Design, Management, Performance

Keywords

Systems, spectrum agility, dynamic spectrum access, cognitive and software-defined radios, cross-layer interactions

1. INTRODUCTION

Spectrum Agility (SA) [1] involves opportunistic utilization of idle spectrum bands, and provides greater communication bandwidth to applications. SA-enabled devices can exploit idle licensed channels instead of getting stuck at crowded and noisy unlicensed bands (e.g., 2.4 GHz ISM band), and hence, increase available bandwidth. Cognitive Radio (CR) or Software-Defined Radio (SDR) [7] is the key enabling technology for SA devices.

In spite of SA's potential to provide a greater overall communication bandwidth, its usefulness to the applications could be very limited in its current form. For instance, SA framework needs to be aware of *spectral opportunities*. Any licensed channel, free from usage by its authorized device (also called the channel's *primary device*) is considered as a spectral opportunity. The awareness of spectral opportunity necessitates frequent sensing or scanning of various channels, during which no application data transmission is possible. This could result in additional delays and reduced bandwidth availability to the application traffic. In addition to sensing, the act of switching to a different channel (for instance, when a primary device is detected on a licensed channel) has associated non-trivial delays, which include the delay of resetting the interface to a new channel as well as additional channel coordination time among communicating entities.

In summary, basic SA operations can adversely affect application performance, which may not be acceptable when a certain level of delay and bandwidth guarantees are needed for applications, e.g., multimedia streaming. We argue that there is a trade-off between SA operations and application requirements, and propose an application-aware adaptive scanning and switching mechanism in order to improve application performance. We call the enhanced SA scheme *Context-Aware Spectrum Agility* (CASA), where *context* comprises application hints as well as current channel conditions. This approach requires a cross-layer interaction mechanism between application and link layers. We have designed a simple and lightweight framework (called *CLIF*) to facilitate the cross-layer exchange. However, we do not focus on CLIF in this paper because of lack of space.

This paper is organized as follows. The related work is discussed in Section 2. CASA is outlined in Section 3, and its preliminary evaluation is presented in Section 4. The paper concludes with Section 5.

2. RELATED WORK

There have been several proposals for SA protocols [3, 5, 10], and research in this area is being actively pursued under FCC directives [6]. Trade-offs between modularity and performance in system design have been discussed in [2] and [9]. Cross-layer exchange leading to application awareness has been proved to be very effective in power management of wireless interfaces [2]. CASA uses a similar approach in order to make SA application-aware.

3. CONTEXT-AWARE SPECTRUM AGILITY

3.1 Overview

The issues associated with SA are an outcome of SA design being oblivious to that actual requirements that must be met. Hence, the key idea behind CASA is to accommodate application needs, given the channel conditions. The goal is to improve the user-perceived application performance by

MobiCom'07, September 9–14, 2007, Montréal, Québec, Canada.
ACM 978-1-59593-681-3/07/0009.

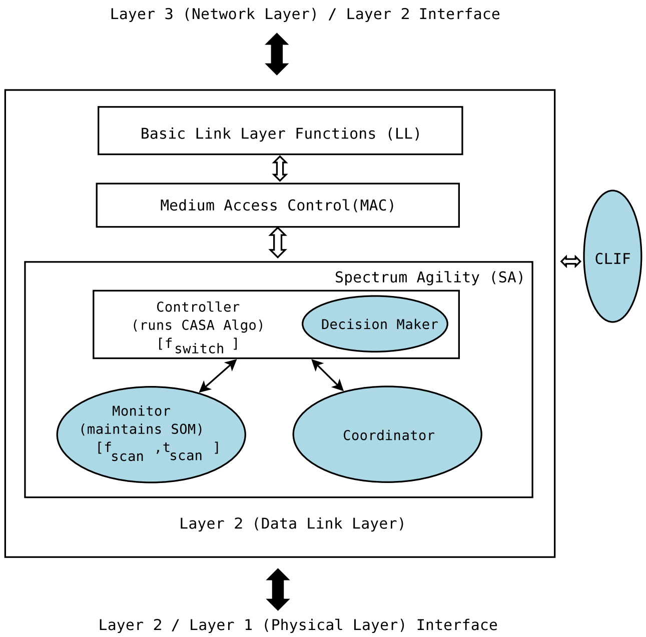

Figure 1: Design of a typical spectrum agility module operating in the data link layer (layer 2). The CASA algorithm augments the "decision-maker" to form "controller".

exposing a small amount of application context, without introducing any significant overhead. We utilize the simple and effective concept of correlating the future with recent past, in estimating SA behavior.

We consider two most important requirements for a network application: *minimum bandwidth* and *maximum tolerable latency*, during a communication session (or a traffic burst). Here, the term "latency" or "delay" denotes the end-to-end communication delay that an application can tolerate. We formalize the notion of *push factor* (representing insufficient bandwidth) that induces a channel switch, and counters the *switch inertia* (representing delay guarantees) of staying at a channel.

The CASA algorithm (running as part of the controller—Figure 1) makes use of the requirement information provided by the applications in order to manage the key SA parameters and channel-switch decisions. Current channel characteristics are also factored into the algorithm. Thus, the *context* comprises information from both the application layer as well as the physical layer.

3.2 The Parameters Involved

Network data transfer by an application occurs in bursts lasting for a variable period of time followed by *quiet* periods. For every upcoming traffic burst, an application exports the expected start time (t_{start}), required bandwidth (b_s), and allowed inter-packet delay (d). There are multiple (say n) network applications running at any given instant, and all the network traffic necessarily passes through the same wireless interface (assuming existence of a single interface). Hence, at the spectrum agility module (SAM) level, only the aggregate of application parameter values are considered, so the actual application-layer inputs to the algorithm are:

- $T_{start} = \min\{t^i_{start}\}, 1 \leq i \leq n$
- $B_s = \sum_{i=1}^{n} b^i_s$
- $D = \min\{d^i\}, 1 \leq i \leq n.$

Based on the available context, the algorithm adjusts two basic spectrum agility parameters: scanning frequency (f_{scan}) and scanning duration (t_{scan}). The actual values of these parameters are chosen from a set of allowed values determined by physical-layer characteristics as well as regulatory policies. CASA introduces adaptive channel switching with a switch-decision-making process running at frequency f_{switch}.

The send bandwidth (or BS) experienced on the current channel is an important input to the algorithm. Also, CASA uses channel utilization values available from the *Spectral Opportunity Map* (SOM) maintained by the sensing module (Figure 1). More specifically, CASA checks if there are channels that have lower utilization—represented by set U—than the current channel, in its decision-making process. Thus, BS and U constitute the physical-layer context available to CASA.

3.3 The CASA Algorithm

The CASA algorithm (Figure 2) involves the execution of a set of decision-making steps. It executes periodically every *epoch* of duration $1/f_{switch}$.

The algorithm initializes the current *push factor*, P_{curr}, of the current epoch to 0, and reads in the delay (D) and bandwidth (B_s) requirements for the next epoch. The cumulative weighted average of previous push factors is represented by P_{past}. In the first two steps, the algorithm adjusts the scanning frequency (f_{scan}) and scanning duration (t_{scan}), in order to meet the delay requirement (D). The total duration for a scan should be at most half the delay requirement. This prevents any violation of the delay requirements while waiting for a scan to finish. Further, the algorithm ensures that the duration between invocation of successive scans is at least D. This approach allows enough time for the held-up and new packets to go through, while allowing SA to be viable with sufficient opportunities for spectrum sensing.

In subsequent steps of the algorithm, the push factor is calculated. This is done by checking the extent to which the requirements are satisfied, and if SA is viable in current circumstances. Any shortfall in meeting a requirement increases the push factor. Past history of push factors (P_{past}) is used to augment P_{curr}. Based on the combined push factor ($P_{curr} + P_{past}$), if there are better channels available, CASA decides in favor of a channel-switch in a probabilistic fashion—the greater the push factor, the better the chances for a switch, and vice-versa. Probabilistic channel switching prevents overcrowding on a single channel, which could happen if multiple secondary devices (or groups) deterministically switch to a channel that is globally perceived to be the best at the moment.

We employ a simple, history-based technique to estimate BS, and also to account for *legacy* application traffic. The details are omitted due to lack of space.

4. EVALUATION

We now present preliminary evaluation results of CASA, but have limited the discussion to a few important observations due to lack of space.

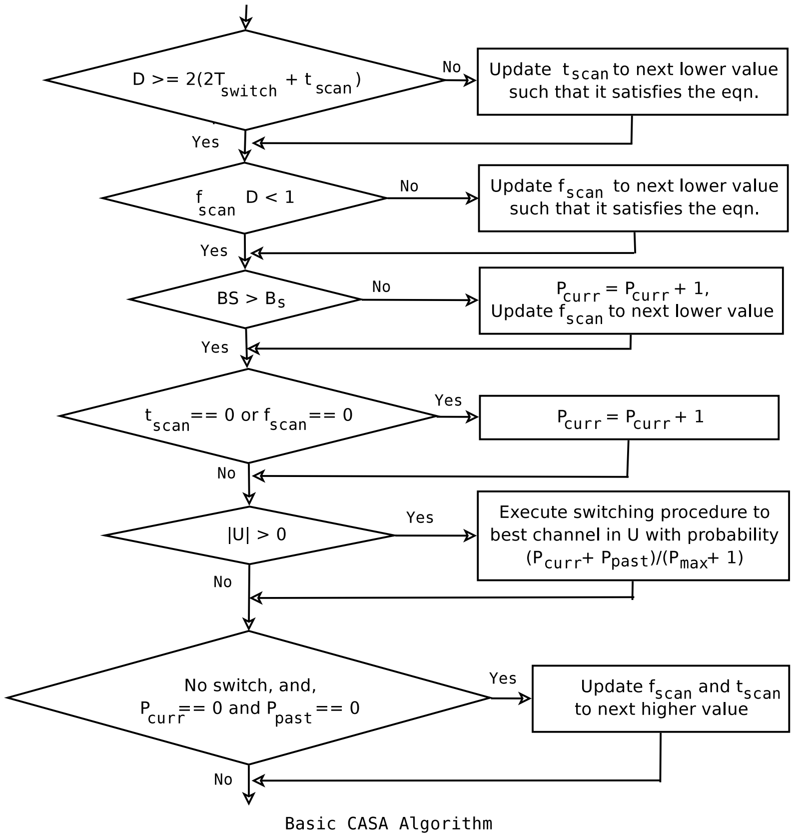

Figure 2: An overview of the basic CASA algorithm. Only key steps executed every epoch are shown.

We implemented SAM (Figure 1) in the *ns-2.29* simulator, as a sublayer of 802.11b MAC protocol. The SA modules are basic in functionality and provide the key operations of scanning, switching and coordination [1, 4]. Note that we are not concerned with details of the protocols for each of these operations. Our aim is to see if context-awareness in SA leads to better network application performance over SA with no context knowledge, irrespective of the underlying details of how SA is implemented. We use the term "traditional or conventional SA" to denote a SA protocol that does not utilize CASA.

4.1 Evaluation Metrics

The following *application-centric* metrics are used to evaluate CASA.

- Overall application throughput received during application run (T).
- Fraction of application run-time during which the send bandwidth requirement is satisfied (F_b).
- Fraction of application run-time during which the delay requirement is satisfied (F_d).

There are the following two important overheads introduced by CASA.

- Sub-optimal channel selection and switches (CS). This could lead to poor SA behavior.
- Cross-layer communication delay for each burst of traffic (CD).

The overhead CD is equivalent to the delay in executing a simple system call. This delay is found to be 1 μs on average, and hence CD is negligible, as the frequency of such calls is in the order of several seconds.

CS is introduced when the CASA algorithm operates, primarily because CASA adjusts scanning frequency (f_{scan}) in response to expected delay requirements. A reduction in f_{scan} leads to stale information about channel states in SOM, and may lead to poor SA decisions.

4.2 Evaluation Setup

In ns-2 simulation scenarios, there are 4 licensed channels. Each channel contains a primary group consisting of two nodes (sender-receiver pair). We use random ON/OFF UDP traffic to simulate application behavior, with average burst time and idle time ranging from 3 to 6 seconds. During active traffic, the sender in the primary group sends packets of 500 bytes each at rates varying between 400 and 800 kbps. There are 3 secondary groups in each simulation scenario. Two of the secondary groups start communication on the same channel, while the remaining one secondary group starts on a different channel. The traffic for a secondary group is also a random ON/OFF process like that of primary groups. The delay requirement (D) is different for each traffic burst and is a uniform random variable between

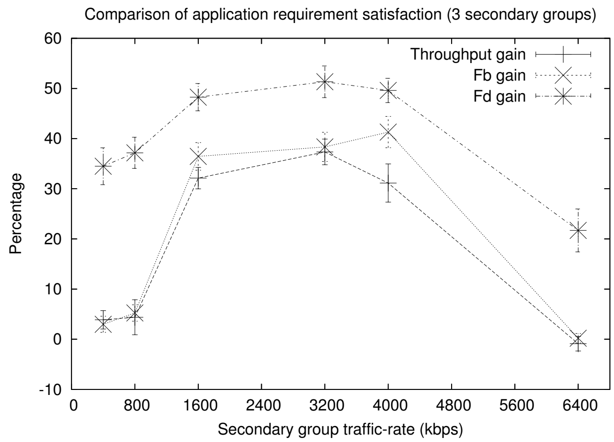

Figure 3: Performance comparisons for a secondary group, in a shared multi-group environment.

10 to 100ms. The delay requirement range mirrors the typical requirement of delay-/bandwidth-sensitive applications (e.g., video/audio streaming web-cast). We vary the application traffic rate and delay, and record the values of the performance metrics. For each scenario, 20 simulation runs (each of 300s) are executed and the performance metric values are averaged.

For simulating traditional SA, we use $f_{scan} = 1$s, and $t_{scan} = 0.025$s. We use the following values of the CASA parameters.

1. $f_{scan} \in \{2, 1, 0.5, 0.25, 0\}$ (per second).
2. $t_{scan} \in \{0.05, 0.025, 0.0125, 0\}$ (second).
3. $f_{switch} = 1$ (per second). Epoch length is $1/f_{switch}$.

The rationale behind choosing these values for f_{scan} and t_{scan} can be found in [4, 8].

4.3 Results and Discussion

CASA is shown to outperform traditional SA on throughput metric (Figure 3), especially when requirements are more stringent. One of the reasons for this is adaptive scanning. CASA cuts down on f_{scan} and t_{scan} when, for example, the required bandwidth is higher than what can be sustained with current scanning frequency at the given physical-layer capacity. The gains with CASA is also due to adaptive channel-switches. Such channel-switches spread out the secondary groups throughout the spectrum, ensuring better chances for data transmission. On the contrary, traditional SA suffers from channel degradation due to inefficient spectrum sharing, especially when application traffic-rate requirement exceeds the effective channel capacity. The improvement in throughput is approximately 35%.

Apart from throughput gains, Figure 3 shows that CASA also performs better on other performance metrics. Note that improvements in F_b mirror the trend of gains in T, which is expected. The results show that CASA increases the resilience of SA protocols in supporting application demands to a much greater degree.

5. CONCLUDING REMARKS

We argued for and showed the importance of application-awareness for spectrum agility. Fundamental operations involved in spectrum agility (e.g., channel sensing and switching) can cause application-traffic interruptions, and adversely influence bandwidth- and/or delay-sensitive applications. We proposed a systems-based optimization mechanism to address this issue. It relies on the significant correlation of immediate future with past observations in channel conditions and application behaviors. Combining application-awareness and channel-state knowledge with spectrum agility, we proposed an improved spectrum agility scheme called CASA. Our preliminary evaluation has shown CASA to increase the resilience of the SA protocol in supporting higher bandwidth and stringent delay requirements. We plan to analyze CASA's performance on realistic traffic patterns, and investigate the security implications of the proposed architecture. We also plan to extend CASA to operate on devices with multiple wireless interfaces.

Acknowledgments

The work described in this paper was supported in part by NSF under Grant CNS-0519498, and Intel Corp.

6. REFERENCES

[1] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty. Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey. *Computer Networks Journal (Elsevier)*, 50:2127–2159, September 2006.

[2] M. Anand, E. B. Nightingale, and J. Flinn. Ghosts in the machine: Interfaces for better power management. In *The Second Annual International Conference on Mobile Systems, Applications and Service (MOBISYS)*, 2004.

[3] S. Banerjee, A. Mishra, V. Brik, V. Shrivastava, and V. Bahl. Towards an architecture for efficient spectrum slicing. In *Eighth IEEE Workshop on Mobile Computing Systems and Applications (HotMobile)*, February 2007.

[4] C.-T. Chou, H. Kim, K. G. Shin, and S. S. N. What and how much to gain by spectral agility? *IEEE Journal on Selected Areas in Communications (JSAC)*, 25(3), April 2007.

[5] C. Cordeiro, K. Challapali, and S. Shankar. IEEE 802.22: The first worldwide wireless standard based on cognitive radios. In *IEEE International Symposium on New Frontiers in Dynamic Spectrum Access (DySPAN)*, November 2002.

[6] FCC. *Spectrum Policy Task Force (SPTF) Report*, 2002. ET Docket No. 02-135.

[7] S. Haykin. Cognitive Radio: Brain-empowered wireless communications. *IEEE Journal on Selected Areas in Communications (JSAC)*, 23(2):201–220, February 2005.

[8] H. Kim, C. Cordeiro, K. Challapali, and K. G. Shin. An experimental approach to spectrum sensing in cognitive radio networks with off-the-shelf IEEE 802.11 devices. In *IEEE Consumer Communications and Networking Conference (CCNC)*, 2007.

[9] B. Lampson. Hints for computer system design. In *9th Symposium on Operating Systems Principles (SOSP)*, pages 33–48, 1983.

[10] L. Ma, X. Han, and C. C. Shen. Dynamic open spectrum sharing MAC protocol for wireless ad hoc networks. In *IEEE DySPAN*, pages 203–213, November 2005.