This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2015.2478458, IEEE Transactions on Mobile Computing

Toward Optimal Distributed Monitoring of Multi-Channel Wireless Networks

Dong-Hoon Shin, *Member, IEEE,* Saurabh Bagchi, *Senior Member, IEEE,* and Chih-Chun Wang, *Senior Member, IEEE*

Abstract—This paper studies an optimal channel assignment problem for passive monitoring in multi-channel wireless networks, where a set of sniffers capture and analyze the network traffic to monitor the wireless network. The objective of this problem is to maximize the total amount of traffic captured by sniffers by judiciously assigning the radios of sniffers to a set of channels. This problem is NP-hard, with the computational complexity growing exponentially with the number of sniffers. We develop *distributed online* solutions for large-scale and dynamic networks. The dynamism in the network may arise from mobility of the nodes being monitored. Our algorithm is guaranteed to achieve at least $1 - \frac{1}{e}$ times the optimum, regardless of the network topology and the channel assignment of nodes to be monitored, while providing a distributed solution amenable to online implementation. Further, our algorithm is cost-effective, in terms of communication and computational overheads, due to the use of purely local communication and the incremental adaptation to network changes. We present two operational modes of our algorithm for two types of networks that change at different rates; one is a proactive mode for fast-varying networks, while the other is a reactive mode for slowly-varying networks. Simulation results demonstrate the effectiveness of the two modes of our algorithm and compare it to the theoretically optimal algorithm.

Index Terms—Wireless networks, multi-channel, monitoring, distributed algorithm, approximation algorithm.

1 Introduction

Thanks to the explosive growth of wireless communications and networking technologies in the last decade, these technologies not only have become essential to people's daily lives, but also are being deeply embedded into physical infrastructure systems. Such a tight integration of advanced wireless technologies into physical systems can potentially benefit a variety of applications and areas, including energy, healthcare, transportation and defense systems. Nevertheless, the key to success lies in the reliability and security of these systems. An essential ingredient for achieving high reliability and security is the high-quality monitoring of the underlying wireless communications.

Passive monitoring is a widely-used and effective technique to monitor wireless networks. In this, a set of sniffers (i.e., software or hardware devices that intercept and log packets) are used to capture and analyze network traffic between other nodes, in order to estimate network conditions and performance. Such estimates are utilized for efficient network operation, such as network resource management, network configuration, fault detection/diagnosis and network intrusion detection.

Over the past few years, the use of multiple channels in wireless networks, especially in Wireless Mesh Networks (WMNs), have been extensively studied (e.g., [1]–[3]). It is known that equipping nodes with multiple radios, each tuned to one of multiple orthogonal bands (or channels), can significantly increase the capacity of the network. On the other hand, utilizing multiple channels in wireless networks brings up a challenging issue with passive monitoring: how to assign a set of channels to sniffers' radios in order to accomplish the given monitoring objective, e.g., capturing as large an amount of traffic, or covering as large a number of nodes, as possible. This problem arises because monitoring resources, i.e., the number of sniffers and the number of radios that each sniffer has, are limited and thus it may not be feasible to monitor traffic on all channels continuously. Therefore, the channel selections for sniffers' radios should be judiciously coordinated in order to accomplish the given monitoring objective.

The sniffer-channel assignment problem for passive monitoring in multi-channel wireless networks has received increasing attention in recent years. The existing works [4]–[20] have studied the problem, with different formulations and different perspectives. The focus of this paper is on *distributed* solutions that are amenable to *online* implementation. There are a number of important reasons for the need of such distributed solutions.

- The solution has to be distributed because networks can be large and centralized solutions are not scalable.
- The solution should not rely on a single powerful entity—which has a high computational power, a large memory and no significant energy constraint—for its entire operation. This is because

- *D.-H. Shin is with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85282, USA. E-mail: donghoon.shin.2@asu.edu.*
- *S. Bagchi and C.-C. Wang are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47906, USA. E-mail: {sbagchi, chihw}@purdue.edu.*

This work was supported in part by NSF grants ECCS-1407603 and CCF-1422997.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2015.2478458, IEEE Transactions on Mobile Computing

not only such solutions relying on a single entity are vulnerable to a single-point failure, but also there are often cases where deploying such a powerful entity is infeasible, e.g., in ad hoc wireless networks.

- The solution has to restrict the information exchange among nodes in their neighboring regions, in order to lower the communication overhead.

In this paper, we develop *distributed* and *online* solutions for large-scale and dynamic networks. Specifically, we formulate an *Optimal Sniffer-Channel Assignment* (OSCA) problem, where the objective is to maximize the *monitoring coverage*, which is defined as the total weight of the nodes monitored by the sniffers. Since OSCA is NP-hard, we focus on approximate solutions that can be computed in polynomial time and design a *distributed* algorithm for OSCA, termed DA-OSCA. It is a Linear Program (LP) rounding algorithm—which first solves an LP relaxation of OSCA and then converts the *fractional* solution to an integer feasible solution to OSCA. We show that DA-OSCA achieves at least $1-\frac{1}{e}$ of the optimal monitoring coverage, regardless of the network topology and the channel assignment of nodes to be monitored. Further, DA-OSCA attains this performance guarantee while requiring only *local* communication among neighboring sniffers and adapting *incrementally* to network changes. In addition, we devise two operational modes of DA-OSCA, thereby allowing it to adapt more efficiently to network changes at two different rates. One is a proactive mode, which is applicable to networks that change at a fast rate. The other is a reactive mode, which allows DA-OSCA to reassign the channels to sniffers only if the monitoring coverage is not high enough. This mode is more suitable for networks that vary at a slow rate. Our simulation results demonstrate the efficacy of these two modes of DA-OSCA.

The main technical contributions of this paper are summarized as follows:

1) We design a *fully distributed* algorithm to solve the LP relaxation of OSCA, termed DA-LP$_{\text{OSCA}}$. It is based on the Proximal Optimization Algorithm (POA) [21] combined with a dual approach. DA-LP$_{\text{OSCA}}$ is a *cost-effective* distributed algorithm, compared to the standard POA [21]. Specifically, the standard POA requires a two-level convergence structure, which is not suitable for distributed algorithms due to high computational and communication overhead. On the other hand, without such an inefficient convergence structure, DA-LP$_{\text{OSCA}}$ can still converge to the optimal solution.

2) We design a *fully distributed* rounding algorithm, *Opportunistic Channel Assignment Algorithm* (OCAA), which achieves at least $1-\frac{1}{e}$ ($\approx$ 0.632) of the maximum monitoring coverage. To this end, we first develop a centralized rounding algorithm for OSCA based on the pipage rounding technique in [22]. We then design OCAA by using a metric called *coverage improvement*, derived from the centralized rounding algorithm, which guides sniffers to make good decisions on their channel selection.

3) We develop a duality-based information aggregation procedure, used in the reactive mode of DA-OSCA, to efficiently estimate the monitoring coverage of a given sniffer-channel assignment. Such an estimation of monitoring quality is needed to determine the followings: i) whether DA-OSCA needs to be invoked in order to improve the degraded monitoring coverage due to the changes in network condition; ii) when to terminate DA-LP$_{\text{OSCA}}$ and round the fractional solution, i.e., whether the fractional solution at an iteration of DA-LP$_{\text{OSCA}}$ is sufficiently close to the optimal (fractional) solution.

The rest of the paper is organized as follows. Section 2 reviews existing works related to this paper. Section 3 describes the problem formulation, discusses the hardness of OSCA, and presents a summary of the proposed distributed algorithm. Sections 4 and 5 present and analyze the distributed algorithm for the LP relaxation of OSCA and the distributed rounding algorithm, respectively. Section 7 presents the two operational modes of DA-OSCA. Section 8 shows the simulation results. Finally, Section 9 gives concluding remarks. Due to space limitations, the proofs are provided in a separate supplemental file available on the TMC website.

2 Related Work

The optimal placement of monitoring nodes for monitoring coverage maximization, in *single-channel* wireless networks, has been studied by Subhadrabandhu *et al.* [23]–[25]. The work [23] studies the problem of how to select an optimal subset of monitoring nodes to execute Intrusion Detection Modules (IDSs), given a budget on the number of monitoring nodes to be used. The goal is to maximize the number of nodes covered (i.e., monitored) by the selected monitoring nodes. The work [24] allows for IDSs that may periodically stop functioning due to operational failure or compromise by intruders. It develops a framework to counter the failure of IDSs, and studies the problem of how to find a minimum set of monitoring nodes to execute IDSs, while covering all nodes in the network. The work [25] allows for IDSs that may periodically fail to detect attacks and generate false alarms, and develops a similar framework to that of [24]. In all of these works [23]–[25], nodes are assumed to use a single common channel, and thus there is no issue of channel assignment for monitoring nodes.

The sniffer-channel assignment problem in *multi-channel* wireless networks has been studied by the works [4]–[20], with different problem formulations and different perspectives. The works [4]–[11] have studied OSCA, its variant, or a generalized problem. Our prior works [4], [5] have studied a more generalized problem than OSCA, i.e., how to optimally place sniffers and assign their channels to monitor multi-channel WMNs, assuming stationary networks. Chhetri *et al.* [6], [7] have studied OSCA (i.e., the MEC problem in [6], [7]) for two models of sniffers, assuming different capabilities

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2015.2478458, IEEE Transactions on Mobile Computing

of sniffers in capturing traffic. Our previous work [8] has studied a generalized version of OSCA allowing for imperfect sniffers, where each node must be monitored by a required number of sniffers to ensure an acceptable quality of monitoring. Chen *et al.* [9] have studied the sniffer-channel selection problem for monitoring Wireless Local Area Networks (WLANs), formulating the two optimization problems: how to minimize the maximum number of channels that a sniffer listens to; how to minimize the total number of channels that the sniffers listen to. The recent works [10], [11] have studied the sniffer-channel selection problem, with the goal to maximize the quality of monitoring. Du *et al.* [10] presented a Monte Carlo enhanced Particle Swarm Optimization (MC-PSO) algorithm, while Xia *et al.* [11] proposed a Multiple Quantum Immune Clone Algorithm (MQICA).

Complementary to the works above, there have been studies [12]–[14] on trade-offs between assigning the radios of sniffers to channels known to be busiest based on the current knowledge, versus exploring channels that are under observed. Arora *et al.* [12] proposed two policies that sequentially learn the user activities while making decisions on the sniffer-channel assignment. A drawback of the two sequential learning policies in [12] is high computational costs due to the NP-hardness of the decision problem. Hence, Zheng *et al.* [13], [14] presented two *approximate* online learning algorithms that are computationally efficient. In the works [15], [16], Hassanzadeh *et al.* proposed a taxonomy to categorize existing solutions for intrusion detection in WMNs. In [15], they investigated the attack-and-fault tolerance of IDS. In [16], they studied two classes of monitoring techniques for intrusion detection in WMN, namely, traffic agnostic and resourceful, and traffic aware and resourceful. Zeng *et al.* [17] proposed a measurement architecture using distributed sniffers for delay monitoring in wireless sensor networks, and studied a sniffer placement problem for efficient delay measurement.

The aforementioned works [4]–[11], which studied OSCA, its variant, or a generalized problem, focus on *centralized* algorithms. In contrast, the works [18], [19] and our earlier work [20], upon which this paper builds, presented *distributed* algorithms to solve OSCA. While the work [14] also presents a distributed algorithm, it is under a different setting, i.e., for online learning. A major difference between the works [18], [19] and this paper is the very different approaches to solve the problem, in terms of the trade-off between the optimality of the solution and the time complexity of algorithm. Specifically, our proposed distributed algorithm, DA-OSCA, which is based on the LP rounding approach, is a *polynomial-time* algorithm that guarantees an approximation ratio of $1 - \frac{1}{e}$. On the other hand, the distributed algorithms in [18], [19], based on a Gibbs sampler approach, guarantees the optimality of the solution but may not converge in polynomial time. We would like to point out that our approach, which sacrifices the optimality for time efficiency, is more suitable for distributed algorithms that need to be agile to the changes of network. Besides, this paper addresses the practical issue of how to efficiently adjust the sniffer-channel assignment as the network changes, which is not handled in [18], [19], by presenting the two operational modes of DA-OSCA.

3 Problem Formulation

Consider a wireless network with a set N of nodes to be monitored. Each node's radio is tuned to a wireless channel chosen from a set C of available wireless channels with $|C| \geq 2$. Each node $n \in N$ is assigned a non-negative weight w_n. These weights of nodes can be used to capture various application-specific objectives of monitoring. For example, one can use the weights to capture data rates of nodes. In this scenario, we would assign higher weights to the nodes transmitting larger volumes of data, thereby biasing our algorithm to monitor such nodes more. Or, for security monitoring, one can assign the weights by taking into account the nodes' trustworthiness computed based on the previous monitoring results. Here, a node that has been found to be compromised before (and repaired thereafter) will be assigned a higher weight.

We are given a set S of sniffers, each of which has to determine a wireless channel from C to tune its radio to. We say that a sniffer and a node are *neighbors* if the sniffer can overhear the node, and also that two sniffers are *neighbors* if both of the sniffers can overhear a node (by tuning their radios to the same channel as the node). We denote the set of the neighboring sniffers of sniffer s by $N(s)$, and the set of the neighboring nodes of sniffer s by $L(s)$. We say that a node is *covered* if the node is overheard by at least one sniffer being tuned to the same channel as the node. We are given a collection of *coverage-sets*, $\mathcal{K} := \{K_{s,c} \subseteq N \colon s \in S,\, c \in C\}$, where a coverage-set $K_{s,c}$ includes the nodes that can be covered by sniffer s being tuned to channel c. We define a *group* $\mathcal{K}_s := \{K_{s,c} \colon c \in C\}$ to denote the collection of all coverage-sets of sniffer s over all channels. We define *monitoring coverage* to be the total weight of the covered nodes. Our objective is to maximize the monitoring coverage by judiciously choosing one coverage-set from each group[1]. Here, the *group budget constraint*, i.e., only one coverage-set can be chosen from each group, arises since each sniffer has only one radio and the radio can be tuned to only one channel at a time. We refer to the optimization problem described above as the ***Optimal Sniffer-Channel Assignment*** (OSCA) problem.

In OSCA, for ease of exposition, we assumed that all nodes and all sniffers have only one radio. However, the multi-radio case, i.e., the case when nodes and sniffers are equipped with multiple radios, can be easily cast into OSCA by regarding each radio of a node (or sniffer) as a different node (or sniffer) with a single radio. In the

[1] In cases where a node is covered by two or more adjacent sniffers tuned on the same channel, the sniffers can communicate with each other and pick the one to monitor the node for efficiency.

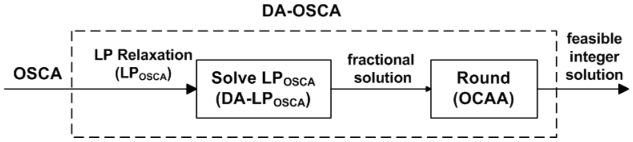


Fig. 1. Distributed Algorithm for OSCA (DA-OSCA).

multi-radio case, one might think that OSCA requires an additional constraint to ensure that each sniffer tune its radios to a set of distinct channels in C, for the efficient use of its radio resources. However, note that tuning two radios of a sniffer to the same channel means choosing the same coverage-sets, say $K_{s,c}$, twice, which results in a lower coverage than that achieved by choosing two distinct coverage-sets including $K_{s,c}$. Therefore, in the multi-radio case, any efficient algorithm would always choose a set of distinct channels in C for the multiple radios of each sniffer, without any additional constraint.

TABLE 1
Summary of Notations

Notation	Definition
n / N	Index/Set of nodes
s / S	Index/Set of sniffers
c / C	Index/Set of wireless channels
$N(s)$ / $L(s)$	Set of the neighboring sniffers/nodes of sniffer s
$K_{s,c}$	Coverage-set: Set of nodes covered by sniffer s operating on channel c
w_n	Weight assigned to node n
x_n / $y_{s,c}$	0/1 variable to indicate if node n is covered (or coverage-set $K_{s,c}$ is chosen) by a solution
d	A parameter of DA-LP$_{\text{OSCA}}$, defined in Eq. (6)
p_n	Dual variable assigned to node n, defined in Eq. (7)
β	A parameter of DA-LP$_{\text{OSCA}}$, defined in Eq. (9)
I	Number of inner-level iterations in DA-LP$_{\text{OSCA}}$
B_1	Maximum number of nodes covered by any sniffer operating on any channel
B_2	Maximum number of neighboring sniffers to any node

3.1 Hardness of OSCA

The following two theorems on the hardness of OSCA have been proved in [6].

***Theorem** 1:* OSCA is NP-hard.

This means that the computational complexity to solve OSCA grows exponentially with the number of sniffers, unless $P = NP$. Many target applications of OSCA have more than a handful of sniffers. Also, the problem has to be solved repeatedly (e.g., whenever the channel assignment of nodes changes) at runtime. Therefore, this theorem points us toward finding approximate solutions that will be applicable to practical networks.

***Theorem** 2:* For any $\epsilon > 0$, it is NP-hard to approximate OSCA within a factor of $\frac{7}{8} + \epsilon$ of the optimum.

In other words, the best achievable approximation ratio for OSCA is at most $\frac{7}{8}$.

3.2 Summary of Proposed Distributed Algorithm

We design a distributed approximation algorithm to solve OSCA, referred to as DA-OSCA, which guarantees to achieve at least $1 - \frac{1}{e}$ (≈ 0.632) of the maximum monitoring coverage. DA-OSCA solves the following Integer Linear Program (ILP) of OSCA, denoted by ILP$_{\text{OSCA}}$:

$$\text{maximize} \quad \sum_{n \in N} w_n x_n \tag{1}$$

$$\text{subject to} \quad x_n \leq \sum_{(s,c):\, n \in K_{s,c}} y_{s,c} \qquad \forall n \in N, \tag{2}$$

$$\sum_{c \in C} y_{s,c} \leq 1 \qquad \forall s \in S, \tag{3}$$

$$0 \leq x_n, y_{s,c} \leq 1 \qquad \forall n \in N, s \in S, c \in C, \tag{4}$$

$$x_n, y_{s,c} \in \{0, 1\} \qquad \forall n \in N, s \in S, c \in C. \tag{5}$$

Here, each node $n \in N$ is associated with an indicator variable $x_n \in \{0, 1\}$: $x_n = 1$ indicates that node n is covered by the given solution. Each coverage-set $K_{s,c} \in \mathcal{K}$ is associated with an indicator variable $y_{s,c} \in \{0, 1\}$: $y_{s,c} = 1$ indicates that the radio of sniffer s is tuned to channel c. The objective function (1), together with the constraints (2) and (5), makes $x_n = 1$ if at least one coverage-set containing node n is chosen for a solution.

DA-OSCA is a Linear Program (LP) rounding based algorithm. It consists of two components (see Fig. 1): 1) distributed algorithm (DA-LP$_{\text{OSCA}}$) to solve the LP relaxation of OSCA (i.e., Eqs. (1)–(4)), denoted by LP$_{\text{OSCA}}$; 2) Opportunistic Channel Assignment Algorithm (OCAA) to round the fractional solution yielded by DA-LP$_{\text{OSCA}}$ in a distributed fashion. Intuitively, DA-OSCA first obtains a global knowledge of the optimal solution to OSCA through DA-LP$_{\text{OSCA}}$, and then uses this knowledge to determine the channels for sniffers through OCAA.

For convenience, we summarize in Table 1 the notations frequently used in this paper (some of which are defined later).

4 Distributed Algorithm for LP$_{\text{OSCA}}$

4.1 Proximal Optimization Algorithm for LP$_{\text{OSCA}}$

The basic idea to solve LP$_{\text{OSCA}}$ is based on the *Proximal Optimization Algorithm* (POA) [21, Ch. 3.4.3] combined with a dual approach. We introduce a set of auxiliary variables, $\{x_n^{\text{aux}}, y_{s,c}^{\text{aux}} \colon n \in N, s \in S, c \in C\}$, and transform LP$_{\text{OSCA}}$ into the following equivalent quadratic program, denoted by QP$_{\text{OSCA}}$:

$$\text{maximize} \quad \sum_{n \in N} w_n x_n - \frac{1}{2d}\Big(\sum_{n \in N} (x_n - x_n^{\text{aux}})^2 + \sum_{\forall (s,c)} \big(y_{s,c} - y_{s,c}^{\text{aux}}\big)^2 \Big) \tag{6}$$

subject to Eqs. (2)–(4).

Here, d is a positive constant. The rationale behind this transformation is to resolve an issue that arises when we solve the dual problem of LP$_{\text{OSCA}}$, due to the linearity of the objective function of LP$_{\text{OSCA}}$[2]. By observing that it must follow that $x_n^{\text{aux}} = x_n$ and $y_{s,c}^{\text{aux}} = y_{s,c}$ to maximize the objective function (6), it is easy to verify that QP$_{\text{OSCA}}$ is equivalent to LP$_{\text{OSCA}}$.

The POA to solve QP$_{\text{OSCA}}$, referred to as POA-QP$_{\text{OSCA}}$, proceeds as follows. At the t-th iteration, $t = 1, 2, 3, \ldots,$ POA-QP$_{\text{OSCA}}$ executes the following two steps:

S1: Fixing $\vec{x}^{\text{aux}} = \vec{x}^{\text{aux}}(t)$ and $\vec{y}^{\text{aux}} = \vec{y}^{\text{aux}}(t)$, where $\vec{x}^{\text{aux}}(t)$ and $\vec{y}^{\text{aux}}(t)$ are given from the previous iteration, solve QP$_{\text{OSCA}}$ with respect to $\vec{x}$ and $\vec{y}$. Let the solution be $\vec{x}(t)$ and $\vec{y}(t)$.

S2: Let $\vec{x}^{\text{aux}}(t+1) = \vec{x}(t)$ and $\vec{y}^{\text{aux}}(t+1) = \vec{y}(t)$.

As the number of iterations, t, tends to infinity, the sequence of vectors generated by POA-QP$_{\text{OSCA}}$ with any initial values (i.e., $\vec{x}^{\text{aux}}(1)$ and $\vec{y}^{\text{aux}}(1)$) converges to the optimal solution QP$_{\text{OSCA}}$ [21, Ch. 3.4.3].

4.2 Duality Approach to Step S1 of QP$_{\text{OSCA}}$

Note that, in each iteration of POA-QP$_{\text{OSCA}}$, we have to solve a global optimization problem at Step S1. We will use a dual approach to solve the problem at Step S1. The rationale behind this is that the dual problem has a simple form of constraints and is easily decomposable, thus allowing us to design a distributed algorithm to solve the problem at Step S1.

We derive the dual problem of the optimization problem at Step S1. For notational simplicity, we define $\vec{z} := (\vec{x}, \vec{y})$ and $\vec{z}^{\text{aux}} := (\vec{x}^{\text{aux}}, \vec{y}^{\text{aux}})$, and denote by Z the constraint set of $\vec{z}$ satisfying Eqs. (3) and (4). We define a set of Lagrange Multipliers for the $|N|$ constraints of Eq. (2) as $\vec{p} := (p_n : n \in N)$ and the Lagrangian function of QP$_{\text{OSCA}}$ with the fixed $\vec{x}^{\text{aux}}$ and $\vec{y}^{\text{aux}}$ as the following:

$$L(\vec{z}, \vec{p}; \vec{z}^{\text{aux}}) := \sum_{n \in N} w_n x_n + \sum_{n \in N} p_n \Big(\sum_{(s,c):n \in K_{s,c}} y_{s,c} - x_n \Big) - \frac{1}{2d} \Big(\sum_{n \in N} \left(x_n - x_n^{\text{aux}}\right)^2 + \sum_{\forall (s,c)} \left(y_{s,c} - y_{s,c}^{\text{aux}}\right)^2 \Big). \tag{7}$$

The dual problem is then given by

$$\begin{aligned} &\text{minimize} \quad D(\vec{p}; \vec{z}^{\text{aux}}) := \max_{\vec{z} \in Z} L(\vec{z}, \vec{p}; \vec{z}^{\text{aux}}) \\ &\text{subject to} \quad \vec{p} \geq 0. \end{aligned} \tag{8}$$

Since the dual objective function D is differentiable due to the quadratic terms in Eq. (7), we can now employ the Gradient Projection Algorithm (GPA) [21, Ch. 3.3.2] to solve the dual problem.

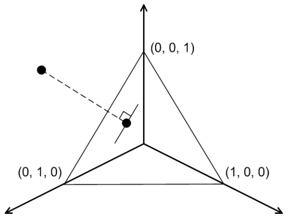


Fig. 2. An illustration of projection $[\cdot]_{Y_i}^{+}$ for $|C| = 3$.

The GPA to solve the dual problem has the following iterations: for $i = 0, 1, 2, \ldots,$

$$p_n(i+1) = \left[p_n(i) + \beta g_n(i)\right]_{[0,+\infty)}^{+}, \tag{9}$$

$$\text{where } g_n(i) := \frac{\partial D}{\partial p_n}\bigg|_{p_n = p_n(i)} = x_n^*(i) - \sum_{(s,c):n \in K_{s,c}} y_{s,c}^*(i).$$

Here, $\beta > 0$ is the step size, $[\vec{p}]_A^{+}$ denotes the projection to a set A, which maps $\vec{p}$ to the point in A that is closest to $\vec{p}$, and $(\vec{x}^*(i), \vec{y}^*(i)) \in Z$ is the optimal solution to the following maximization problem: for given $\vec{p}(i)$,

$$\begin{aligned} &\text{maximize} \quad L(\vec{z}, \vec{p}(i); \vec{z}^{\text{aux}}) \\ &\text{subject to} \quad \vec{z} \in Z. \end{aligned} \tag{10}$$

To solve this problem, we rewrite Eq. (7) as

$$L(\vec{z}, \vec{p}; \vec{z}^{\text{aux}}) = \sum_{n \in N} \Big(-\frac{1}{2d} \left(x_n - x_n^{\text{aux}}\right)^2 + (w_n - p_n) x_n \Big) + \sum_{\forall (s,c)} \Big(-\frac{1}{2d} \left(y_{s,c} - y_{s,c}^{\text{aux}}\right)^2 + y_{s,c} \sum_{n \in K_{s,c}} p_n \Big). \tag{11}$$

Using Eq. (11), we can decompose the problem in Eq. (10) into the following independent subproblems:
1) for each $n \in N$,

$$\begin{aligned} &\text{maximize} \quad -\frac{1}{2d} \left(x_n - x_n^{\text{aux}}\right)^2 + (w_n - p_n(i))\, x_n \\ &\text{subject to} \quad 0 \leq x_n \leq 1 \end{aligned}$$

2) for each $s \in S$,

$$\begin{aligned} &\text{maximize} \quad \sum_{c \in C} \Big(-\frac{1}{2d} \left(y_{s,c} - y_{s,c}^{\text{aux}}\right)^2 + y_{s,c} \sum_{n \in K_{s,c}} p_n(i) \Big) \\ &\text{subject to} \quad \vec{y}_s \in Y_s := \Big\{ \vec{y}_s \colon \sum_{c \in C} y_{s,c} \leq 1, y_{s,c} \geq 0 \ \forall c \Big\}. \end{aligned}$$

Note that each subproblem can be solved independently at each node and at each sniffer *using purely local communication*. By solving each subproblem, we can obtain the following solution:

$$x_n^*(i) = \left[x_n^{\text{aux}} + d(w_n - p_n(i))\right]_{[0,1]}^{+}, \tag{12}$$

$$\vec{y}_s^*(i) = \Big[\Big(y_{s,c}^{\text{aux}} + d \sum_{n \in K_{s,c}} p_n(i) : c \in C \Big) \Big]_{Y_s}^{+}. \tag{13}$$

In Eq. (13), the projection $[\cdot]_{Y_s}^{+}$, illustrated in Fig. 2, can be performed by using Alg. 1. The proof of its correctness can be found in Appendix A in the separate supplemental file.

[2] Specifically, since the objective function (1) of LP$_{\text{OSCA}}$ is linear, it is not strictly concave. As a result, the dual problem of LP$_{\text{OSCA}}$ may not be differentiable at every point. This leads to a difficulty when we use the Gradient Projection Algorithm [21, Ch. 3.3.2] to solve the dual problem. However, such a difficulty will be resolved with QP$_{\text{OSCA}}$, since the objective function of QP$_{\text{OSCA}}$ is strictly concave due to the added quadratic terms and thus is differentiable.

Algorithm 1 Projection Algorithm

```
// The following procedure projects a vector $\vec{v}$ onto $V := \{\vec{v}' : v'_c \geq 0$ for all $c \in C$ and $\sum_{c \in C} v'_c \leq 1\}$
$C' \leftarrow C$
while $|C'| > 0$ and $\sum_{c \in C'} v_c > 1$ do
    $a \leftarrow \frac{1}{|C'|}\left(1 - \sum_{c' \in C'} v_{c'}\right)$    // $a < 0$
    for each $c \in C'$ do
        $v_c \leftarrow v_c + a$
        if $v_c \leq 0$ then
            $v_c \leftarrow 0$
            $C' \leftarrow C' \setminus \{c\}$    // $v_c > 0$ for $c \in C'$
        end if// this makes $\sum_{c \in C} v_c = 1$, since $\sum_{\forall c \notin C'} v_c = 0$
    end for
end while
return $\vec{v}$
```

In summary, we can obtain the solution to the dual problem in Eq. (8) by alternately updating Eq. (9) and Eqs. (12)–(13). As $i \to \infty$, the sequence of vectors given by Eq. (9) converges to the optimal solution of the dual problem [21, Proposition 3.4]. Once the optimal solution of the dual problem is obtained, the optimal solution of the primal problem, i.e., Step S1 of POA-QP$_{\text{OSCA}}$, can be computed with Eqs. (12)–(13) [26, Ch. 5.5.3].

4.3 Distributed Algorithm for LP$_{\text{OSCA}}$

We present in Alg. 2 a formal description of the Distributed Algorithm for LP$_{\text{OSCA}}$ (DA-LP$_{\text{OSCA}}$). Note that DA-LP$_{\text{OSCA}}$ requires only *local communications* among neighboring nodes. In many monitoring applications, it would be desirable that DA-LP$_{\text{OSCA}}$ be run by only sniffers, since DA-LP$_{\text{OSCA}}$ is needed for sniffers to determine their channels. In such cases, one can let one of the neighboring sniffers to node n act as a proxy and update the variables associated with node n (i.e., x_n, x_n^{aux}, p_n) on behalf of node n.

The standard POA [21, Ch. 3.4.3] requires a two-level convergence structure. Specifically, the inner-level iterations (i.e., the `for` loop in Lines 3–8) must converge before the next outer-level iteration (i.e., the `while` loop in Lines 1–11) begins. However, such a two-level convergence structure is not suitable for distributed algorithms, since it requires a mechanism to determine when to stop inner-level iterations. Such a mechanism would not only decrease the convergence speed of DA-LP$_{\text{OSCA}}$ but also incur substantial communication overheads. The rationale behind this intuition is as follows. As the number of inner-level iterations, I, increases, the improvement of the solution quality at each inner-level iteration would decrease. However, such later inner-level iterations that achieve a small improvement would be wasteful, since solving Step S1 of POA-QP$_{\text{OSCA}}$ is only an intermediate step to solve the ultimate problem, i.e., QP$_{\text{OSCA}}$. This intuition is verified by simulation results (Fig. 5).

Algorithm 2 DA-LP$_{\text{OSCA}}$

```
1:  while TRUE do
2:      // Step 1 of POA-QP$_{\text{OSCA}}$
3:      for $i = 0$ to $I$ do
4:          Each node $n$ and each sniffer $s$ compute $x_n(i)$ and $\vec{y}_s(i)$ according to Eqs. (12) and (13), respectively. Then, each sniffer $s$ sends the updated values $\vec{y}_s(i)$ to its neighboring nodes.
5:          if $i \neq I$ then
6:              Each node $n$ computes $p_n(i+1)$ according to Eq. (9), then sends $p_n(i+1)$ to its neighboring nodes and sniffers.
7:          end if
8:      end for
9:      // Step 2 of POA-QP$_{\text{OSCA}}$
10:     Each node $n$ and each sniffer $s$ set initial values of their variables for the next iteration as the following:
            $x_n^{\text{aux}} \leftarrow x_n(I)$ and $p_n(0) \leftarrow p_n(I)$    (node $n$)
            $\vec{y}_s^{\text{aux}} \leftarrow \vec{y}_s(I)$    (sniffer $s$).
11: end while
```

Based on this intuition, we fix the number of inner-level iterations of DA-LP$_{\text{OSCA}}$ to 2 (i.e., $I = 1$), and find a good approximate solution to Step S1 of POA-QP$_{\text{OSCA}}$. In the following theorem, we show that, even with $I = 1$, DA-LP$_{\text{OSCA}}$ still converges to the optimal solution, provided that the step size β (in Eq. (9)) is sufficiently small[3]. The proof is given in Appendix B in the separate supplemental file.

***Theorem** 3:* Let $\vec{z}^{\text{aux},t} = \vec{z}^{\text{aux}}(1)$ and $\vec{p}^{\,t} = \vec{p}(1)$ at the t-th outer-level iteration in DA-LP$_{\text{OSCA}}$. As $t \to \infty$, $\vec{z}^{\text{aux},t}$ and $\vec{p}^{\,t}$ converge to the optimal primal solution and the optimal dual solution of QP$_{\text{OSCA}}$, respectively, if

$$\beta < \frac{1}{2d(B_1+1)\cdot\max\{|C|, B_2+1\}},$$

where $B_1 := \max_{s\in S, c\in C} |K_{s,c}|$ represents the maximum number of nodes covered by a sniffer operating on a channel, and $B_2 := \max_{n\in N} |\{K_{s,c} : n \in K_{s,c}\}|$ represents the maximum number of neighboring sniffers of a node.

Theorem 3 suggests that the value of d (in Eq. (6)) should be set to a smaller value, so that β can be chosen to a larger value, thereby achieving a larger improvement at each inner-level iteration. On the other hand, a smaller value of d will make the objective function of QP$_{\text{OSCA}}$ (Eq. (6)) more deviated from that of the original problem (Eq. (1)). Hence, this would increase the number

[3] Our result in Theorem 3 can be viewed as a parallel version of the improved POA in [27]. This work has previously used the idea of fixing the number of inner-level iterations. However, the results in [27] are based on the assumption that the coefficients in the constraints of the underlying LP problem must be non-negative. Hence, the results in [27] cannot be applied to our problem, LP$_{\text{OSCA}}$, since LP$_{\text{OSCA}}$ has negative coefficients as well in the constraints.

of outer-level iterations, leading to slow convergence of DA-LP$_{\text{OSCA}}$. Our simulation results (Fig. 4) reveal that a smaller value of d leads to slower convergence up to a near-optimal solution (i.e., 95% of the maximum coverage) but a faster convergence to the exact optimal solution.

Note that, in practice, we need a condition to terminate the outer-level iterations and thus Alg. 2. We address this issue in Section 7 by devising two operational modes of DA-LP$_{\text{OSCA}}$ for two types of networks that change at different rates.

5 Distributed Rounding Algorithm

Recall that the solution yielded by DA-LP$_{\text{OSCA}}$ may contain non-integer values, since the integer constraint (Eq. (5)) are relaxed in LP$_{\text{OSCA}}$. Hence, to obtain a feasible solution to ILP$_{\text{OSCA}}$, we have to convert the fractional solution yielded DA-LP$_{\text{OSCA}}$ to an integer solution. Note that given the values of $\{y_{s,c}\}$, ILP$_{\text{OSCA}}$ immediately determines the values of $\{x_n\}$ as $x_n = \min\left\{1, \sum_{(s,c):n\in K_{s,c}} y_{s,c}\right\}$. Hence, we first round the non-integer values of $\{y_{s,c}\}$, and then determine the integer values of $\{x_n\}$ as described above.

We first develop a centralized rounding algorithm for OSCA, called PIPAGE-OSCA, by employing the pipage rounding technique in [22]. We then design a distributed rounding algorithm for OSCA based on PIPAGE-OSCA.

5.1 Pipage Based Centralized Rounding Algorithm

We describe below how PIPAGE-OSCA operates.

PIPAGE-OSCA. It rounds the fractional values of $\vec{y}$, yielded by DA-LP$_{\text{OSCA}}$, in an iterative fashion. At each iteration, it adjusts two non-integer values of a sniffer s, say y_{s,c_1} and y_{s,c_2}, as follows. For the given fractional solution $\vec{y}$, it creates two new (fractional) solution, denoted by $\vec{y}^{(1)}$ and $\vec{y}^{(2)}$. They have the same value as $\vec{y}$ in all entries, except the two entries of indices (s, c_1) and (s, c_2): for $\vec{y}^{(1)}$, $y^{(1)}_{s,c_1} = 0$ and $y^{(1)}_{s,c_2} = y_{s,c_1} + y_{s,c_2}$; for $\vec{y}^{(2)}$, $y^{(2)}_{s,c_1} = y_{s,c_1} + y_{s,c_2}$ and $y^{(2)}_{s,c_2} = 0$. Note that $\vec{y}^{(1)}$ and $\vec{y}^{(2)}$ both include at least one more integer value than $\vec{y}$. Between $\vec{y}^{(1)}$ and $\vec{y}^{(2)}$, it chooses as the new solution, used for the input in the next iteration, the one that achieves a higher value of function $F(\vec{y})$, defined as:

$$F(\vec{y}) := \sum_{n\in N} w_n\Big(1 - \prod_{(s,c):n\in K_{s,c}} (1 - y_{s,c})\Big).$$

It repeats this process until it obtains an integer solution.

PIPAGE-OSCA has the following guarantee. The proof is given in Appendix C in the separate supplemental file.

***Lemma** 1:* Given a (fractional) solution to LP$_{\text{OSCA}}$ that attains a constant factor α of the optimum of LP$_{\text{OSCA}}$, PIPAGE-OSCA yields an integer solution to OSCA that achieves at least $\alpha(1 - \frac{1}{e})$ times the optimum of OSCA.

5.2 Distributed Rounding Algorithm: OCAA

We design a distributed rounding algorithm based on PIPAGE-OSCA. We first define a metric, called *coverage improvement*, which allows sniffers to evaluate the decision criterion for the new solution at each iteration of PIPAGE-OSCA, i.e., whether $F(\vec{y}^{(1)}) \geq F(\vec{y}^{(2)})$, in a distributed manner. Given a set of values $\vec{y}_{N(s)} := \{y_{s',c} \colon s' \in N(s), c \in C\}$, where $N(s)$ denotes the set of neighboring sniffers to sniffer s, we define the *coverage improvement* of coverage-set $K_{s,c}$ as

$$I\left(K_{s,c}; \vec{y}_{N(s)}\right) := \sum_{n\in K_{s,c}} w_n\Big(\prod_{s'\neq s\colon\, n\in K_{s',c}} (1 - y_{s',c})\Big).$$

Intuitively, $I(K_{s,c}; \vec{y}_{N(s)})$ means the *expected* monitoring coverage gain achieved by sniffer s tuning its radio to channel c, when the (fractional) value of $y_{s',c}$ is viewed as the probability that sniffer $s' \in N(s)$ tunes its radio to channel c. Note that each sniffer can compute the coverage improvements over all channels by communicating with only its neighbors.

We have the following lemma. The proof is given in Appendix D in the separate supplemental file.

***Lemma** 2:* $F(\vec{y}^{(1)}) \geq F(\vec{y}^{(2)})$ if $I(K_{s,c_1}, \vec{y}_{N(s)}) \leq I(K_{s,c_2}, \vec{y}_{N(s)})$.

Observe that when PIPAGE-OSCA rounds the fractional values of $\vec{y}_s$, the values of $\vec{y}_{s'}$ for all $s' \neq s$ remain the same. Hence, in the consecutive iterations of PIPAGE-OSCA to round all fractional values of $\vec{y}_s$, the coverage improvements of sniffer s, i.e., $I(K_{s,c}, \vec{y}_{N(s)})$ for all $c \in C$, would not change. This observation and Lemma 2 imply that the consecutive iterations of PIPAGE-OSCA will result in sniffer s being assigned to the channel whose coverage-set achieves the maximum coverage improvement. With this finding, we design a distributed rounding algorithm for OSCA, called the *Opportunistic Channel Assignment Algorithm* (OCAA), described in Alg. 3.

In OCAA, sniffers determine their channel in a sequential manner specified by $\mathcal{P}$. The partition $\mathcal{P}$ can be determined a priori, or through an ad hoc coordination among sniffers, e.g., using one of the existing scheduling algorithms at the Medium Access Control (MAC) layer. In each iteration, the sniffers in $P_i \in \mathcal{P}$ can determine their channel in parallel. Each sniffer s selects the channel c^* that maximizes the coverage improvement, $I(K_{s,c}; \vec{y}_{N(s)})$, based on the given values of $\vec{y}_{N(s)}$ (Line 3). Thereafter, each sniffer s in P_i informs its neighboring sniffers of its decision on the channel (Line 4), so that they can update the values of $\vec{y}_s$ and compute the coverage improvements in later iterations.

To make this clear, we give an illustrative example. Here, we let $S = \{s_1, s_2, s_3, s_4\}$, $C = \{c_1, c_2, c_3\}$, and $\mathcal{P} = \{P_1, P_2\}$, where $P_1 = \{s_1, s_2\}$ and $P_2 = \{s_3, s_4\}$. Suppose that DA-LP$_{\text{OSCA}}$ yielded the following fractional solution: $\vec{y}_{s_1} = (0.2, 0.3, 0.5)$, $\vec{y}_{s_2} = (0.4, 0.3, 0.3)$, $\vec{y}_{s_3} = (0.8, 0.2, 0)$ and $\vec{y}_{s_4} = (0.1, 0.6, 0.3)$. In the first round of

Algorithm 3 Opportunistic Channel Assignment Algorithm (OCAA)

```
1: // $\mathcal{P} := \{P_i\}$ is a partition of $S$ such that no two
   sniffers in any $P_i$ are neighbors.
2: for $i = 1$ to $|\mathcal{P}|$ do
3:     Each sniffer $s \in P_i$ tunes its radio to a channel
       $c^* \in C$ such that:
           $I(K_{s,c^*}; \vec{y}_{N(s)}) = \max_{c \in C} I(K_{s,c}; \vec{y}_{N(s)})$.
       Here, $\vec{y}_{N(s)}$ initially takes the values yielded by
       DA-LP$_{\text{OSCA}}$, and then is updated according to the
       channel selection of sniffer $s'$ neighboring sniffers.
4:     Each sniffer $s$ sends its determination on channel
       selection to its neighboring sniffers.
5: end for
```

OCAA, sniffers s_1 and s_2 round $\vec{y}_{s_1}$ and $\vec{y}_{s_2}$ to an integer solution, say $(0,0,1)$ and $(0,1,0)$, based on the values of $I(K_{s,c}, \vec{y}_{N(s)})$ for s_1, s_2, and $c \in C$. In the second round, upon receiving the decisions made by s_1 and s_2, sniffers s_3 and s_4 update the values of $I(K_{s,c}, \vec{y}_{N(s)})$ for s_3, s_4, and $c \in C$, and use them to round $\vec{y}_{s_3}$ and $\vec{y}_{s_4}$ to an integer solution, say $(1,0,0)$ and $(0,1,0)$.

Since OCAA is an distributed implementation of PIPAGE-OSCA, OCAA maintains the same performance as PIPAGE-OSCA. We thus have the following theorem.

***Theorem** 4:* Given a (fractional) solution to LP$_{\text{OSCA}}$ that attains a constant factor α of the optimum of LP$_{\text{OSCA}}$, OCAA yields an integer solution to OSCA that achieves at least $\alpha(1 - \frac{1}{e})$ times the optimum of OSCA.

Here, the factor α comes from an approximate solution yielded by LP$_{\text{OSCA}}$. Note that we can make α arbitrarily close to 1 by sufficiently increasing the number of outer-level iterations of DA-LP$_{\text{OSCA}}$. We thus have the following corollary.

***Corollary** 1:* DA-OSCA always achieves at least $1 - \frac{1}{e}$ times the maximum coverage of OSCA, regardless of the network topology and the channel assignment of nodes.

6 Scalability and Asynchronous Operation of DA-OSCA

In this section, we first examine the scalability of DA-OSCA, and then discuss how to operate DA-OSCA in an asynchronous fashion.

6.1 Scalability

To examine the scalability of DA-OSCA, we analyze the computational complexity of DA-OSCA per each sniffer. First, in DA-LP$_{\text{OSCA}}$ with $I = 1$, each sniffer iteratively computes Eqs. (9), (12) (on behalf of some of its neighboring nodes) and (13), which involve simple arithmetic operations except $[\cdot]^+_{Y_s}$ in Eq. (13). To compute Eq. (13), each sniffer invokes Alg. 1, which has complexity of $O(|C|^2)$. Hence, in DA-LP$_{\text{OSCA}}$ with $I = 1$, the computational complexity on each sniffer is $O(|C|^2 k)$, where k is the number of the outer-level iterations that DA-LP$_{\text{OSCA}}$ takes to converge. Next, in OCAA, each sniffer s computes the coverage improvement (i.e., $I(K_{s,c}; \vec{y}_{N(s)})$) $|C|$ times. Since the complexity of computing the coverage improvement is $O(B_1 B_2)$ (refer to Theorem 3 for the definition of B_1 and B_2), the computational complexity on each sniffer in OCAA is $O(|C| B_1 B_2)$. Thus, the overall computational complexity on each sniffer in DA-OSCA is $O\left(|C|(B_1 B_2 + k|C|)\right)$. Typically, B_1 and B_2, which are determined by the network topology, would increase at a much lower rate than that at which the network size grows. Also, our simulation results (Figs. 4 and 6) empirically show that k does not grow as the network size increases. Rather, k depends on the difficulty of each instance of OSCA, determined by the setting of several input parameters (see the discussion for Fig. 6 in Section 8.1). These observations point out that DA-OSCA is scalable. Further, we devise two operational modes of DA-OSCA in the next section, which enable DA-OSCA to incrementally adapt to network changes. This significantly reduces k as demonstrated through simulations (see Figs. 11 and 12).

6.2 Asynchronous Operation

We discuss how to operate DA-OSCA in an asynchronous fashion, so as to facilitate its operation in a realistic environment where all sniffers may not be perfectly synchronized. Recall that DA-LP$_{\text{OSCA}}$ converges to the optimal solution with any initial values. This means that even if some sniffers may not update their variables at a few iterations, e.g., due to unreliable links with their neighboring sniffers, DA-LP$_{\text{OSCA}}$ would eventually converge to the optimal solution, albeit through some fluctuations. Hence, DA-LP$_{\text{OSCA}}$ can be executed in an asynchronous manner: if a sniffer receives no update from some of its neighboring sniffers, it proceeds to update its variables without waiting for the update. In this way, DA-LP$_{\text{OSCA}}$ can facilitate its operation coping with large communication delays and temporary link failures. In OCAA, while sniffers determine their channel in a sequential manner, specified by $\mathcal{P}$ (see Line 1 of Alg. 3), each sniffer in $P_i \in \mathcal{P}$ can determine its channel, independent of the other sniffers' decisions in P_i. In this way, we can facilitate the deployment of DA-OSCA while still maintaining its performance.

7 Online Implementation of DA-OSCA

In this section, we describe how to implement DA-OSCA in an online fashion, so that DA-OSCA is agile and adapts incrementally to network changes, such as the changes of channels/weights assigned to nodes and the changes of network topology due to nodes' mobility and sniffers' arrivals/departures. Note that failures and recoveries of nodes and sniffers can also be regarded as departures and arrivals of them, respectively. We first describe the procedure that sniffers need to perform when they find arrivals/departures of their neighboring nodes/sniffers. We then present two operational modes of DA-OSCA, which allow DA-OSCA to adapt more efficiently to network changes at the two different rates.

7.1 Basic Information Update

When sniffer s finds the arrivals or departures of its neighboring nodes, it first updates its coverage-sets (i.e., $\{K_{s,c}\colon c \in C\}$). If a node n arrives, sniffer s, which acts as the proxy for node n (for updating the values of the node n's variables), introduces a set of variables for node n, and then sets their initial values as follows: $x_n = 1$ if node n is covered, and $x_n = 0$ otherwise; $x_n^{\text{aux}} = x_n$; $p_n = 0$. If node n leaves, sniffer s removes the node n's variables.

When a new sniffer s arrives, it first creates its coverage-sets and its variables, and then sets their initial values as follows: $y_{s,c^*} = 1$ for $c^* \in C$ such that K_{s,c^*} achieves the maximum coverage improvement (i.e., $c^* = \text{argmax}_{c \in C}\, I(K_{s,c}; \vec{y}_{N(s)})$) and $y_{s,c} = 0$ for all $c \in C \,/\, \{c^*\}$; $\vec{y}_s^{\text{aux}} = \vec{y}_s$. When sniffer s leaves, one of its neighboring sniffers takes over the proxy duty of sniffer s.

7.2 Mode-I: DA-OSCA for Fast Varying Networks

We present in Alg. 4 the operation of DA-OSCA in Mode-I, where DA-OSCA operates *proactively* to adapt to rapid changes in network condition. The rationale behind this proactive mode is that, when the network changes rapidly, it is cost-effective to run DA-OSCA continuously, rather than running it in a reactive manner. This is because reactive operation of DA-OSCA requires an additional mechanism (like the one in Mode-II) to estimate how good monitoring coverage a sniffer-channel assignment achieves, in order to determine when to start and when to terminate DA-OSCA. Such a coverage-estimation mechanism, however, would require network-wide communication to aggregate the information needed for the coverage estimation. Therefore, a reactive operation of DA-OSCA would require frequent network-wide communication, which is costly, when the network condition changes rapidly.

In Alg. 4, DA-OSCA executes one outer-level iteration of DA-LP$_{\text{OSCA}}$ every T_1 time units (Line 2), and invokes OCAA every lT_1 time units, i.e., every l outer-level iterations of DA-LP$_{\text{OSCA}}$ (Line 4). That is, DA-OSCA keeps updating the knowledge of the optimal solution through DA-LP$_{\text{OSCA}}$, based on which it periodically changes sniffers' channel assignment through OCAA.

Algorithm 4 DA-OSCA in Mode-I

```
1: if $t = t'T_1$, $t' = 1, 2, \cdots$ then
2:   Perform one outer-level iteration of DA-LP$_{\text{OSCA}}$
     (i.e., Lines 3–11 of Alg. 2)
3:   if $t = t''(lT_1)$, $t'' = 1, 2, \cdots$ then
4:     Invoke OCAA
5:   end if
6: end if
```

7.3 Mode-II: DA-OSCA for Slow Varying Networks

In this mode, DA-OSCA operates *on demand*, i.e., only when the channel assignment of sniffers needs to be changed, thus improving the monitoring coverage degraded due to the changes in network condition. For this reactive operation, DA-OSCA requires a mechanism to estimate the monitoring coverage in order to determine: i) whether the invocation of DA-OSCA is needed; ii) whether the number of iterations of DA-LP$_{\text{OSCA}}$ executed is sufficiently large to yield a good approximation solution to LP$_{\text{OSCA}}$. Hence, we first develop a procedure to estimate the monitoring coverage. We then describe how DA-OSCA makes use of the coverage-estimation procedure in the reactive mode.

7.3.1 *Efficient information aggregation procedure to estimate the quality of monitoring coverage*

We present in Alg. 5 an efficient information aggregation procedure to estimate the monitoring coverage. It determines whether the gap between the current monitoring coverage and the maximum monitoring coverage, which is defined as the ratio of the former to the latter, is above a desired level specified by a pre-determined value γ. To estimate the gap, it computes the current monitoring coverage (C_R), and the dual objective function value (D_R). The rationale behind this is that, by the duality theory [26, Ch. 5.1.3], any dual objective function value is an upper bound on the primal optimal value, i.e., the maximum monitoring coverage. To compute C_R and D_R, it aggregates the values of C_s and D_s through the spanning tree of sniffers (Line 2). Then, it verifies that the current monitoring coverage is above the desired level, by checking if $C_R \geq \gamma D_R$ (Line 3). Finally, the evaluation result is sent to all the sniffers through the spanning tree (Line 4). An illustration of the information flow in Alg. 5 is given in Fig. 3.

Alg. 5 has the following performance guarantee. The proof is given in Appendix E in the separate supplemental file.

***Theorem** 5:* If $C_R \geq \gamma D_R$, then $C_R \geq \gamma F_{\text{LP}}^*$, where F_{LP}^* denotes the optimum of LP$_{\text{OSCA}}$.

7.3.2 *Description of Mode-II*

We present in Alg. 6 the operation of DA-OSCA in Mode-II. In this mode, DA-OSCA evaluates the monitoring coverage every T_2 time units, by invoking Alg. 5 (Line 2). If the estimate (r_{MC}) of the gap between the current monitoring coverage and the maximum monitoring coverage is above the desired level, specified by γ_1, DA-OSCA immediately terminates. Otherwise, DA-OSCA starts to solve the new instance of OSCA resulting from the change in network condition (Lines 3–6). To this end, DA-OSCA runs N_o outer-level iterations of DA-LP$_{\text{OSCA}}$,

Algorithm 5 An efficient information aggregation procedure to estimate the quality of monitoring coverage

1: // A pre-constructed spanning tree of sniffers is assumed.

2: **Aggregation of information.** This step is initiated by leaf sniffers, and is executed sequentially along the levels of the spanning tree upwards to the child nodes of the root sniffer. At a level of the spanning tree, sniffer s computes:

$$C_s := \sum_{s' \in \mathrm{CS}(s)} C_{s'} + \sum_{n \in L(s)} w_n \min\Big\{1, \sum_{(s,c):n \in K_{s,c}} y_{s,c}\Big\},$$

$$D_s := \sum_{s' \in \mathrm{CS}(s)} D_{s'} + \sum_{n \in K_{s,c^*}} p_n + \sum_{n \in L(s)} [w_n - p_n]^+,$$

where $c^* \in \mathrm{argmax}_{c \in C} \sum_{n \in K_{s,c}} p_n$, $[x]^+ = \max\{x, 0\}$, and $\mathrm{CS}(s)$ and $L(s)$ denote the set of the child sniffers of sniffer s and the set of the neighboring nodes of sniffer s, respectively. Then, sniffer s sends C_s and D_s to its parent sniffer.

3: **Determination of solution quality.** The root sniffer, denoted by R, computes C_R and D_R as described above, and determines that the current channel assignment achieves the desired monitoring coverage if $C_R \geq \gamma D_R$. Then, the root sniffer R sends its child sniffers a message to inform this determination.

4: **Distribution of determination.** The determination made by the root sniffer is delivered to all sniffers along the spanning tree.

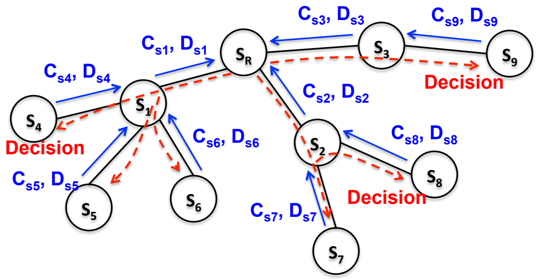


Fig. 3. An illustration of the information flow in Alg. 5.

and repeats it unless the quality of the solution of DA-LP$_{\text{OSCA}}$, denoted by r_{LP}, is sufficiently close to that of the optimal solution of LP$_{\text{OSCA}}$ (Lines 3–5). Evaluating the quality of the solution of DA-LP$_{\text{OSCA}}$ is done by invoking Alg. 5 with a pre-specified precision γ_2. It should be noted that the value of N_o gives a trade-off between the cost of checking the stopping criterion and the cost of running more outer-level iterations of DA-LP$_{\text{OSCA}}$ than required to reach the solution quality. Therefore, the value of N_o needs to be carefully chosen taking into account the convergence speed of DA-LP$_{\text{OSCA}}$. Once a near-optimal solution to LP$_{\text{OSCA}}$ is obtained, DA-OSCA terminates DA-LP$_{\text{OSCA}}$ and then converts the fractional solution to an integer solution to OSCA, through OCAA (Line 6).

Algorithm 6 DA-OSCA in Mode-II

```
1: if $t = t'T_2$, $t' = 1, 2, \cdots$ then
2:   if $r_{\text{MC}} \leq \gamma_1$ (checked by invoking Alg. 5) then
3:     while $r_{\text{LP}} \leq \gamma_2$ (checked by invoking Alg. 5) do
4:       Perform $N_o$ outer-level iterations of DA-LP$_{\text{OSCA}}$ (i.e., Lines 3–11 of Alg. 2)
5:     end while
6:     Invoke OCAA
7:   end if
8: end if
```

8 Simulation

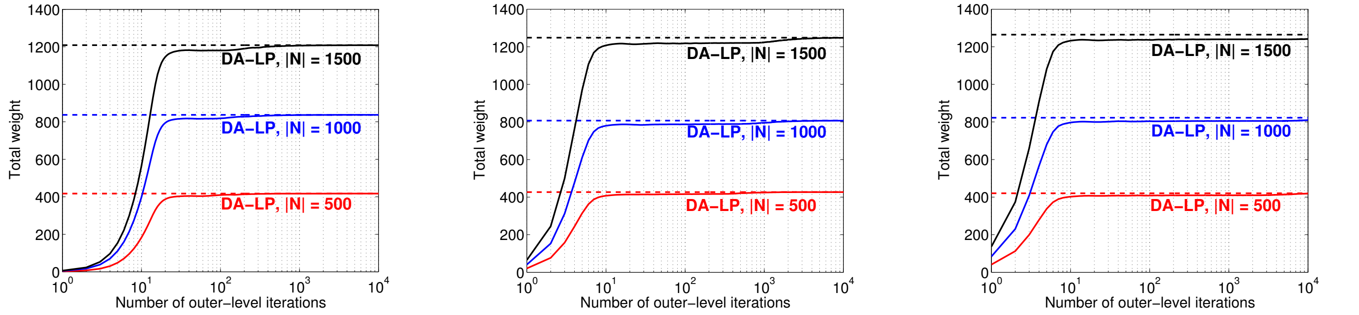
We evaluate the performance of DA-OSCA through simulations. We perform simulations in two kinds of networks: random networks and scale-free networks. In random networks, nodes and sniffers are randomly deployed in a 1×1 square area with a uniform distribution. In scale-free networks, nodes are deployed such that the probability $f(\delta)$ of a node with degree δ follows a power law of the form of δ^{-r}, i.e., the number of nodes with high degree decreases exponentially. In scale-free networks, the nodes with highest degrees are chosen as sniffers, thereby achieving higher coverage. The rationale behind choosing these two kinds of networks is that the performance of DA-OSCA will largely depend on the distribution of the degrees of nodes, which differs significantly in the two kinds of networks.

The basic settings of the network and the parameters of DA-OSCA are as follows, unless specified otherwise. We set $|N|$, $|S|$ and $|C|$ to 500, 50 and 3, respectively. All nodes have an identical weight of one. Each node's radio is tuned randomly to one of the channels in C. In random networks, the receiving range of sniffers is set to 0.15. In scale-free networks, the parameter r of the distribution $f(\delta) = O(\delta^{-r})$ is chosen as $2 < r < 3$. The parameters of DA-OSCA are set as the following: $I = 1$, $d = 0.5$, and β is set according to Theorem 3. In all simulations, the results are the averages over a number of iterations.

8.1 Evaluation of DA-OSCA

Figure 4 shows how the monitoring coverage achieved by DA-LP$_{\text{OSCA}}$ evolves in random networks, for different values of d and $|N|$, as the number of outer-level iterations of DA-LP$_{\text{OSCA}}$ increases. We present only the results for random networks since similar results are observed in scale-free networks. We discover a trend that a smaller value of d leads to a slower convergence in initial iterations but a faster convergence in later iterations. That is, the value of d gives a trade-off between the initial and the overall convergence rates. We also observe that the overall convergence rate becomes slower as the number of nodes increases.

Figure 5 shows the number of inner-level iterations that DA-LP$_{\text{OSCA}}$ takes to attain at least θ times the optimum of LP$_{\text{OSCA}}$, as I increases. We observe that, for all values of θ, the number of inner-level iterations

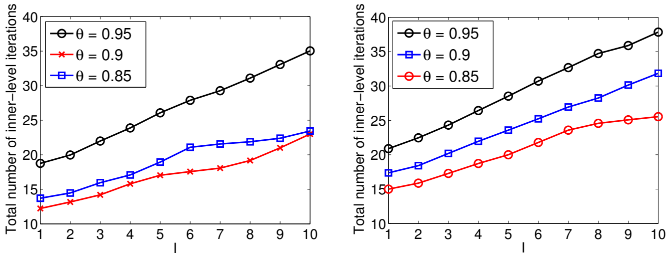


(a) $d = 0.05$

(b) $d = 0.5$

(c) $d = 5$

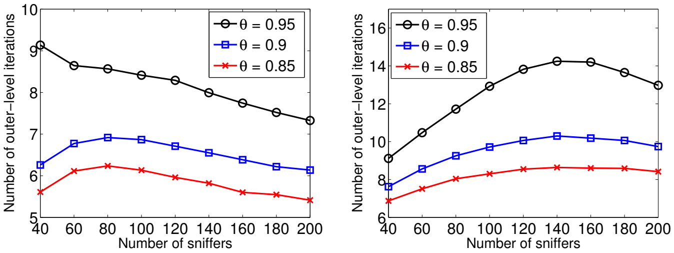
Fig. 4. Evolution of monitoring coverage achieved by DA-LP$_{\text{OSCA}}$ in random networks.



(a) Random network

(b) Scale-free network

Fig. 5. Number of inner-level iterations that DA-LP$_{\text{OSCA}}$ takes to achieve at least a factor θ of the LP$_{\text{OSCA}}$ optimum.



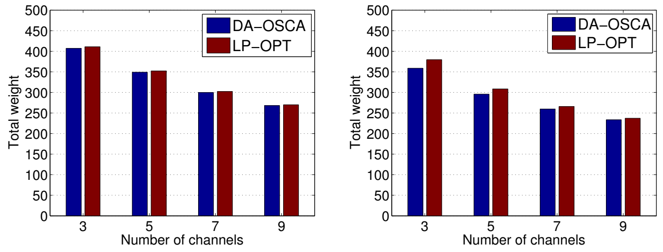
(a) Random network

(b) Scale-free network

Fig. 6. Number of outer-level iterations that DA-LP$_{\text{OSCA}}$ takes to achieve at least a factor θ of the LP$_{\text{OSCA}}$ optimum.

increases as I grows. This agrees with our intuition that a smaller value of I leads to a faster convergence rate of DA-LP$_{\text{OSCA}}$, as discussed in Section 4.3.

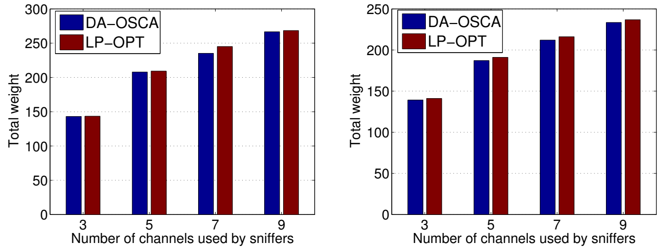
Figure 6 shows the number of outer-level iterations that DA-LP$_{\text{OSCA}}$ takes to attain at least θ times the optimum of LP$_{\text{OSCA}}$, as the number of sniffer increases. We observe that the number of outer-level iterations does not grow with the number of sniffers (and neither with number of nodes, up to $\theta = 0.95$, as observed in Fig. 4). We note that DA-LP$_{\text{OSCA}}$ takes more number of outer-level iterations in scale-free networks than in random networks, for all values of θ, which implies that random networks generate more favorable inputs to DA-OSCA than scale-free networks (which can also be observed in Fig. 9). From this, we see that the convergence of DA-LP$_{\text{OSCA}}$ to a near-optimal solution mainly depends on the difficulty of the instance of OSCA.



(a) Random network

(b) Scale-free network

Fig. 7. Monitoring coverage achieved by DA-OSCA for $|C| = 3, 5, 7, 9$. LP-OPT denotes the optimum of LP$_{\text{OSCA}}$.



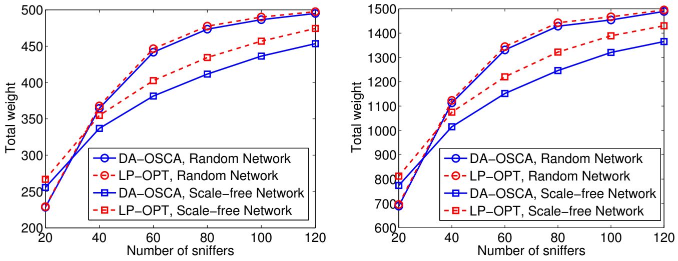
(a) Random network

(b) Scale-free network

Fig. 8. Monitoring coverage achieved by DA-OSCA for $|C|$ = 9: the number of channels used by sniffers increases from 3 to 9.

Figure 7 shows the monitoring coverage of DA-OSCA for different values of $|C|$. We compare the results with the optimal value of LP$_{\text{OSCA}}$, which is used as an upper bound on the maximum monitoring coverage. We see that the monitoring coverage achieved by DA-OSCA is comparable to the maximum coverage for all the values of C and for the both kinds of networks. We also observe that the monitoring coverage decreases as the number of channels grows. This is because as the number of channels increases, each sniffer would cover a smaller number of nodes on each channel.

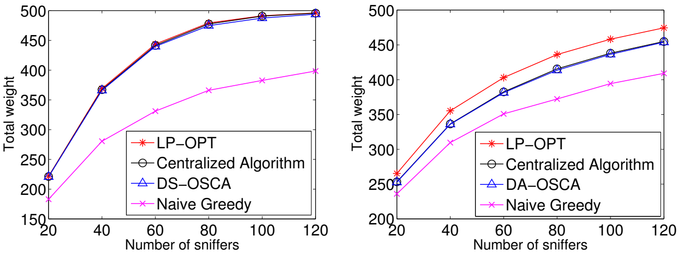
Figure 8 shows the monitoring coverage of DA-OSCA for $|C| = 9$. In this simulation, we restrict the number of channels used by sniffers, which increases from 3 to 9. We see that the monitoring coverage achieved by DA-OSCA is comparable to the maximum coverage. We



(a) Identical assignment

(b) Different assignment

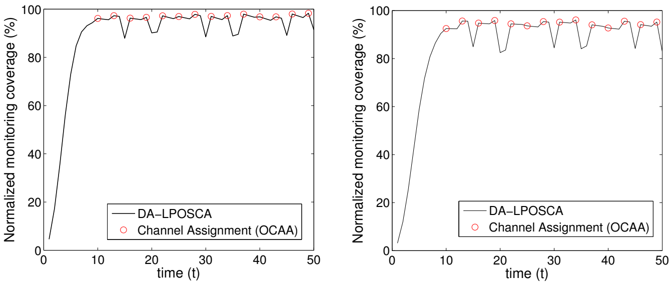
Fig. 9. Monitoring coverage achieved by DA-OSCA for two different weight assignments: (a) all weights have an identical value of 1; (b) the weight of each node is assigned randomly to one of the integers $\{1, 2, 3, 4, 5\}$.



(a) Random network

(b) Scale-free network

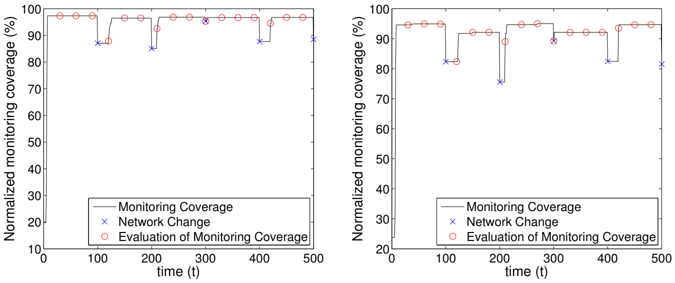
Fig. 10. Comparison of monitoring coverage achieved by DA-OSCA, the centralized algorithm in [5], and a naive greedy algorithm.



(a) Random network

(b) Scale-free network

Fig. 11. Evolution of coverage as DA-OSCA in Mode-I operates proactively in fast-varying networks.



(a) Random network

(b) Scale-free network

Fig. 12. Evolution of coverage as DA-OSCA in Mode-II operates on demand in slowly-varying networks.

also observe that the monitoring coverage increases as the number of channels used by sniffers grows. This is expected since a larger number of channels available for sniffers allows them to choose from more channels, thereby improving the monitoring coverage.

Figure 9 shows the coverage of DA-OSCA for two different weight assignments. We observe similar trends for both the weight assignments and for the both kinds of networks. It is noticeable that DA-OSCA achieves a higher coverage in random networks than in scale-free networks. This is, possibly, because in random networks sniffers are uniformly deployed and thus sniffers have a better topological coverage than in scale-free networks. Also, we observe that the gap between the coverage of DA-OSCA and LP-OPT is smaller in random networks. This again implies that random networks generate more favorable inputs to DA-OSCA than scale-free networks.

Figure 10 shows the comparison of monitoring coverage achieved by DA-OSCA, the centralized algorithm in [5] applied for OSCA, and a naive greedy algorithm that assigns each sniffer to the busiest channel it senses. We observe that DA-OSCA achieves almost the same monitoring coverage as the centralized algorithm in [5]. We also see that DA-OSCA achieves significantly higher monitoring coverage than the naive greedy algorithm, especially in random networks. It is noticeable that the naive greedy algorithm achieves better performance in scale-free networks. This is expected since a significant portion of nodes is assigned to a single channel in scale-free networks.

8.2 Evaluation of Two Modes of DA-OSCA

We now demonstrate the efficacy of the two operational modes of DA-OSCA by evaluating how monitoring coverage evolves as DA-OSCA adapts to the changes of channels assigned to nodes. The channel of each node is assigned randomly to channel 1, 2, or 3 with probabilities 0.2, 0.3, and 0.5, respectively. The channel assignment of a fraction of nodes (randomly chosen between 10% and 40%) changes every 5 time units and every 100 time units in the fast-varying and the slowly-varying networks, respectively. Here, one time unit is defined as the time that it takes to invoke one outer-level iteration of DA-LP$_{\text{OSCA}}$. We set $T_1 = 1$ and $l = 3$ in Mode-I, and $T_2 = 30$, $\gamma_1 = 0.8$, $\gamma_2 = 0.8$, and $N_o = 1$ in Mode-II.

Figure 11 demonstrates Mode-I of DA-OSCA. Here, the monitoring coverage is normalized by the optimal value of LP$_{\text{OSCA}}$. In this simulation, we let DA-OSCA adjust the channel assignment of sniffers 10 time units after the simulation begins. For the both kinds of networks, we observe that the (fractional) coverage achieved by DA-LP$_{\text{OSCA}}$ converges rapidly (within 10 time units) until it reaches about 90% of the maximum coverage, and it flattens out after it goes above 90% of the maximum coverage. We also observe that DA-LP$_{\text{OSCA}}$ quickly recovers the degraded monitoring coverage resulted from the changes of channels assigned to nodes. Within only a few time units, the new channel assignment of sniffers by OCAA attains a high monitoring coverage (above 95% of the maximum coverage).

Figure 12 demonstrates Mode-II of DA-OSCA. We observe large intervals of time where the coverage is flat. This means that, through Alg. 5, DA-OSCA determined that the coverage meets the desired level, and thus terminated without any processing, thereby saving unnecessary cost. We notice that as the network changes, the monitoring coverage is degraded (note the dips) but is quickly recovered (always within 20 time units), due to the on-demand invocation of OCAA. Also, we observe higher improvement of monitoring coverage than required (recall that $\gamma_2 = 0.8$) after the execution of DA-OSCA. We can explain this result by the following two facts: first, OCAA often improves the fractional solution, as observed in Fig. 11; second, DA-$\mathrm{LP_{OSCA}}$ may take more number of outer-level iterations than required as Alg. 5 underestimates the monitoring coverage.

9 Conclusion

In this paper, we developed a distributed online algorithm for the optimal sniffer-channel assignment for passive monitoring in multi-channel wireless networks. Our distributed algorithm guarantees to achieve an approximation ratio of $1 - \frac{1}{e}$, regardless of the network topology and the channel assignment of nodes to be monitored. We also devised two operational modes of the proposed algorithm, for cost-effective operation in two types of networks that have different rates of network changes. One is a proactive mode for fast-varying networks, while the other is a reactive mode for slowly-varying networks. Simulation results show that the proposed algorithm achieves comparable performance to that of the optimal solution, and also demonstrate the effectiveness of the two operational modes.

References

[1] M. Alicherry, R. Bhatia, and L. Li, "Joint Channel Assignment and Routing for Throughput Optimization in Multi-radio Wireless Mesh Networks," in *Proc. of ACM MobiCom*, 2005.

[2] X. Lin and S. Rasool, "A Distributed Joint Channel-Assignment, Scheduling and Routing Algorithm for Multi-Channel Ad Hoc Wireless Networks," in *Proc. of IEEE INFOCOM*, 2007.

[3] A. Dhananjay, H. Zhang, J. Li, and L. Subramanian, "Practical, Distributed Channel Assignment and Routing in Dual-radio Mesh Networks," in *Proc. of ACM SIGCOMM*, 2009.

[4] D.-H. Shin and S. Bagchi, "Optimal Monitoring in Multi-Channel Multi-Radio Wireless Mesh Networks," in *Proc. of ACM MobiHoc*, 2009.

[5] ——, "An Optimization Framework for Monitoring Multi-Channel Multi-Radio Wireless Mesh Networks," *Elsevier Ad Hoc Networks*, vol. 11, no. 3, pp. 926–943, May 2013.

[6] A. Chhetri, H. Nguyen, G. Scalosub, and R. Zheng, "On Quality of Monitoring for Multi-Channel Wireless Infrastructure Networks," in *Proc. of ACM MobiHoc*, 2010.

[7] ——, "On Quality of Monitoring for Multichannel Wireless Infrastructure Networks," *IEEE Transaction on Mobile Computing*, vol. 13, no. 3, pp. 664–677, March 2014.

[8] D.-H. Shin, S. Bagchi, and C.-C. Wang, "Toward Optimal Sniffer-Channel Assignment for Reliable Monitoring in Multi-Channel Wireless Networks," in *IEEE Proc. of SECON*, 2013.

[9] X. Chen, Y.-A. Kim, B. Wang, Y. Song, H. Dinh, and G. Chen, "Sniffer channel selection for monitoring wireless LANs," *Elsevier Ad Hoc Networks*, vol. 35, no. 16, pp. 1994–2003, September 2012.

[10] H.-Z. Du, N. Xia, J.-G. Jiang, L.-N. Xu, and R. Zheng, "A Monte Carlo Enhanced PSO Algorithm for Optimal QoM in Multi-Channel Wireless Networks," *Journal of Computer Science and Technology*, vol. 28, no. 3, pp. 553–563, 2013.

[11] N. Xia, L. Xu, and C. Ni, "Optimal QoM in Multichannel Wireless Networks Based on MQICA," *International Journal of Distributed Sensor Networks*, vol. 2013, 2013.

[12] P. Arora, C. Szepesvari, and R. Zheng, "Sequential Learning for Optimal Monitoring of Multi-channel Wireless Networks," in *Proc. of IEEE INFOCOM*, 2011.

[13] R. Zheng, T. Le, and Z. Han, "Approximate Online Learning for Passive Monitoring of Multi-Channel Wireless Networks," in *Proc. of IEEE INFOCOM*, April 2013.

[14] ——, "Approximate Online Learning Algorithms for Optimal Monitoring in Multi-Channel Wireless Networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 2, pp. 1023–1033, February 2014.

[15] A. Hassanzadeh and R. Stoleru, "On the Attack-and-Fault Tolerance of Intrusion Detection Systems in Wireless Mesh Networks," *Lecture Notes in Computer Science*, vol. 8487, pp. 387–401, 2014.

[16] A. Hassanzadeh, A. Altaweel, and R. Stoleru, "Traffic-and-Resource-Aware Intrusion Detection in Wireless Mesh Networks," *Elsevier Ad Hoc Networks*, vol. 21, pp. 18–41, October 2014.

[17] W. Zeng, J. Cote, X. Chen, Y.-A. Kim, W. Wei, K. Suh, B. Wang, and Z. J. Shi, "Delay Monitoring for Wireless Sensor Networks: An Architecture Using Air Sniffers," *Elsevier Ad Hoc Networks*, vol. 13, pp. 549–559, 2014.

[18] P. Arora, N. Xia, and R. Zheng, "A Gibbs Sampler Approach for Optimal Distributed Monitoring of Multi-Channel Wireless Networks," in *Proc. of IEEE GLOBECOM*, 2011.

[19] Y. Wang, R. Zheng, and Q. Wang, "Self-tuned Distributed Monitoring of Multi-channel Wireless Networks Using Gibbs Sampler," *Elsevier Computer Networks*, vol. 64, 2014.

[20] D.-H. Shin, S. Bagchi, and C.-C. Wang, "Distributed Online Channel Assignment Toward Optimal Monitoring in Multi-Channel Wireless Networks," in *IEEE INFOCOM, Mini-conference*, 2012.

[21] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, New Jersey, 1989.

[22] A. Ageev and M. Sviridenko, "A New Method of Constructing Algorithms with Proven Performance Guarantee," *Jour. of Combinatorial Optimization*, 2004.

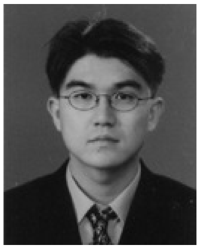
[23] D. Subhadrabandhu, S. Sarkar, and F. Anjum, "A Framework for Misuse Detection in Ad Hoc Networks—Part I," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 274–289, February 2006.

[24] ——, "A Framework for Misuse Detection in Ad Hoc Networks—Part II," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 290–304, February 2006.

[25] ——, "A Statistical Framework for Intrusion Detection in Ad Hoc Networks," in *Proc. of INFOCOM*, April 2006.

[26] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[27] X. Lin and N. B. Shroff, "Utility Maximization for Communication Networks with Multi-path Routing," *IEEE Transactions on Automatic Control*, 2006.



Dong-Hoon Shin is an Assistant Research Professor in the School of Electrical, Computer and Energy Engineering at Arizona State University, Tempe, Arizona, where he was a Postdoctoral Scholar from Aug. 2012 to Aug. 2013. He received his B.E. degree from Korea University, Seoul, South Korea, in 2003, his M.S. degree from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2006, and his Ph.D. degree from Purdue University, West Lafayette, Indiana, in 2012. He has been a graduate intern in Intel's Wireless Standards and Advanced Technologies Group from Aug. 2011 to Jun. 2012. His research interests span the areas of communication, cyber-physical and social networks, with emphasis on the security and privacy of these networks.

Saurabh Bagchi is a Professor in the School of Electrical and Computer Engineering and in the Department of Computer Science at Purdue University, West Lafayette, Indiana. He is an ACM Distinguished Scientist and a senior member of IEEE and ACM. At Purdue, he is the Assistant Director of CERIAS, the security center, an IMPACT Faculty Fellow, and leads the Dependable Computing Systems Laboratory (DCSL) where his group performs research in practical system design and implementation of dependable distributed systems. He has been a Visiting Scientist with IBM's Austin Research Lab since 2011.

Chih-Chun Wang is currently an Associate Professor of the School of Electrical and Computer Engineering of Purdue University. He received the B.E. degree in E.E. from National Taiwan University, Taipei, Taiwan in 1999, the M.S. degree in E.E., the Ph.D. degree in E.E. from Princeton University in 2002 and 2005, respectively. He worked in Comtrend Corporation, Taipei, Taiwan, as a design engineer in 2000 and spent the summer of 2004 with Flarion Technologies, New Jersey. In 2005, he held a postdoctoral researcher position in the Department of Electrical Engineering of Princeton University. He joined Purdue University as an Assistant Professor in 2006, and became an Associate Professor in 2012. He is currently a senior member of IEEE and has been an associate editor of IEEE Transactions on Information Theory since 2014. His current research interests are in the graph-theoretic and algorithmic analysis of iterative decoding and of network coding. Other research interests of his fall in the general areas of networking, optimal control, information theory, detection theory, and coding theory.

Dr. Wang received the National Science Foundation Faculty Early Career Development (CAREER) Award in 2009.