

1-11-2012

Distributed Online Channel Assignment Toward Optimal Monitoring in Multi-Channel Wireless Networks

Dong-Hoon Shin

Purdue University, shin39@purdue.edu

Saurabh Bagchi

Purdue University, sbagchi@purdue.edu

Chih-Chun Wang

Purdue University - Main Campus, chihw@purdue.edu

Follow this and additional works at: <http://docs.lib.purdue.edu/ecetr>

Shin, Dong-Hoon; Bagchi, Saurabh; and Wang, Chih-Chun, "Distributed Online Channel Assignment Toward Optimal Monitoring in Multi-Channel Wireless Networks" (2012). *ECE Technical Reports*. Paper 427.

<http://docs.lib.purdue.edu/ecetr/427>

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.

Distributed Online Channel Assignment Toward
Optimal Monitoring in Multi-Channel Wireless Networks

Dong-Hoon Shih

Saurabh Bagchi

Chih-Chun Wang

TR-ECE-12-01

January 11, 2012

School of Electrical and Computer Engineering

1285 Electrical Engineering Building

Purdue University

West Lafayette, IN 47907-1285

Distributed Online Channel Assignment Toward Optimal Monitoring in Multi-Channel Wireless Networks

Dong-Hoon Shin, Saurabh Bagchi, and Chih-Chun Wang
School of Electrical and Computer Engineering, Purdue University
465 Northwestern Avenue, West Lafayette, IN 47907, USA
Email: {shin39, sbagchi, chihw}@purdue.edu

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 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 to this problem for large-scale and dynamic networks. Prior works have attained constant factor of $1 - \frac{1}{e}$ of the maximum monitoring coverage in a centralized setting. Our algorithm preserves the same ratio while providing a distributed solution that is amenable to online implementation. Also, our algorithm is cost-effective, in terms of communication and computational overheads, due to the use of only local communication and the adaptation to incremental network changes. We present two operational modes of our algorithm for 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 demonstrate the effectiveness of the two modes of our algorithm.

I. INTRODUCTION

We consider a channel assignment problem for passive monitoring in multi-channel wireless networks. Passive monitoring is a widely-used and effective technique to monitor wireless networks, where 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 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. Recently, it has been extensively studied to use multiple channels in wireless networks, especially in wireless mesh networks (WMNs) [1]–[5]. It has been shown that equipping nodes with multiple radios tuned to different non-overlapping channels can significantly increase the capacity of the network. In multi-channel wireless networks, a major challenge with passive monitoring is how to assign a set of channels to each sniffer’s radios such that as large an amount of traffic or large a number of nodes as possible are captured. We call this the *optimal sniffer-channel assignment* (OSCA) problem.

Previous works [6]–[8] have studied variants of OSCA in different perspectives. In our prior work [6], we have

studied a problem of how to optimally place sniffers and assign their channels to monitor multi-channel WMNs, assuming stationary networks. Chhetri *et al.* [7] have studied two models of sniffers that assume different capabilities of sniffers’ capturing traffic. The first, called *user-centric* model, assumes that frame-level information can be captured so that activities of different users are distinguishable. The second, called *sniffer-centric* model, assumes only binary information regarding channel activities, i.e., whether some user is active in a specific channel near a sniffer. In both of the works [6], [7], the authors assume that a prior knowledge of the topology and the channel usages of nodes to be monitored is given to, or can be inferred by, sniffers. On the other hand, Arora *et al.* [8] have studied a trade-off between assigning the radios of sniffers to channels known to be busiest based on the current knowledge, versus exploring channels that are under observed. In addition, Subhadrabandhu *et al.* [9]–[11] have studied a problem of how to optimally place a set of intrusion detection modules for misuse detection in *single-channel* wireless networks.

One can obtain a good approximate solution to OSCA, which is an NP-hard problem (see Section II-B), by extending algorithms in [6], [7]. The work [7] studies a special case of OSCA, where each sniffer has only one radio, while our prior work [6] studies a generalized version of OSCA, i.e., the optimal selection of sniffers and their channels. But, the algorithms in [6], [7] are centralized and offline algorithms. That is, the algorithms requires a central authority that first gathers, from all sniffers, either a prior knowledge of the network topology and the channel usages of all nodes to be monitored [6], or primitive information to estimate the prior knowledge [7], then runs the algorithm and distributes the solution to all sniffers.

These centralized algorithms are not suitable for large-scale and dynamic networks, due to several reasons. The centralized algorithms require an efficient and cost-effective two-way global communication mechanism between the central authority and all sniffers, i.e., the communications from all sniffers to the central authority for the delivery of the prior knowledge, and also the communication from the central authority to all sniffers for the distribution of the solution. However, this is difficult to achieve in large-scale networks, especially in multi-hop wireless networks. Also, such a two-

way global communication needs to be achieved without too much delay, otherwise the centralized algorithms are not agile to frequent network changes, such as channel-usage changes of nodes and network topology changes due to mobility of nodes and arrivals/departures of sniffers. In addition, the centralized algorithms are difficult to deploy in ad hoc wireless networks, which lack the central authority or a powerful node that has a high computational power, a large memory, and no significant energy constraint. Moreover, the powerful node needs to be fault-tolerant or easily replaceable when it fails, since otherwise the entire monitoring system may fail due to a single-point failure.

In this paper, we develop *distributed* and *online* solutions to OSCA for large-scale and dynamic networks. Our distributed algorithm, called DA-OSCA, achieves a *provably good* performance. DA-OSCA can always achieve at least $1 - \frac{1}{e}$ (≈ 0.632) of the maximum monitoring coverage, regardless of the network topology and the channel assignment of nodes to be monitored. Previously, the centralized algorithms in [6], [7] have also attained the ratio $1 - \frac{1}{e}$. However, our DA-OSCA preserves the same ratio while providing a distributed solution that is amenable to online implementation. Also, DA-OSCA is *cost-effective*, in terms of communication and computational overheads, since DA-OSCA requires only local communication among neighboring nodes and also adapts incrementally to network changes. DA-OSCA solves OSCA in two steps. At the first step, DA-OSCA solves distributedly an LP relaxation of OSCA, which is obtained by removing the integer constraints from integer linear program (ILP) formulation of OSCA. At the second step, DA-OSCA rounds distributedly the fractional solution of the LP relaxation to an integer solution, while obtaining a feasible solution to the original ILP. Moreover, the decentralized and adaptive structure of DA-OSCA allows us to operate DA-OSCA in two different modes that are suitable for fast-varying and slow-varying networks, respectively. Specifically, one is a proactive mode for fast-varying network, while the other is a reactive mode for slow-varying networks. With these two operational modes, DA-OSCA can adapt to two different rates of network changes in a cost-effective manner. To demonstrate the effectiveness of DA-OSCA in these modes, we conduct simulations in two kinds of network—random networks and scale-free networks.

The rest of the paper is organized as follows. Section II presents the problem formulation and existing results. Section III presents the distributed algorithm. Section IV describes the online implementation of the distributed algorithm. Section V discusses notes. Section VI presents simulation results. Section VII concludes this paper and discusses future works.

II. PROBLEM FORMULATION

A. Optimal Sniffer-Channel Assignment (OSCA) Problem

We are given a set N of nodes to be monitored, and each node $n \in N$ is tuned to a wireless channel chosen from a set C of available wireless channels, where $|C| \geq 2$. The channels are chosen according to one of many available channel assignment algorithms (e.g., [3], [4], [12]). Each node

n is given 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 transmission rates of nodes. In this scenario, we would assign higher weights to the nodes that transmit 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 nodes' trustworthiness computed based on 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 needs 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 there exists a node that can be overheard by both the sniffers. 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, $\mathcal{K} = \{K_{s,c} \subseteq N : s \in S, c \in C\}$, where a *coverage-set* $K_{s,c}$ contains the nodes that can be covered by sniffer s being tuned to channel c . We define a *group* as a collection of coverage-sets of a sniffer over all channels, i.e. $\mathcal{K}_s = \{K_{s,c} : c \in C\}$. Our objective is to maximize the total weight of the nodes covered by judiciously choosing one coverage-set from each group. Here, the constraint that only one coverage-set can be chosen from each group arises since each sniffer can tune its radio to only one channel at a time, since it has a single radio. We call this constraint the *group budget constraint*, and refer to the optimization problem as the *optimal sniffer-channel assignment* (OSCA) problem.

For ease of exposition, we assume that all of the nodes and the sniffers have only one radio. However, the multi-radio case, where nodes and sniffers are equipped with multiple radios, can be easily mapped to this single-radio case. (Refer to Section V)

B. Hardness of OSCA

We present existing results on the hardness of OSCA.

Theorem 1 (Theorem 1 [7]): OSCA is NP-hard.

This means that the computational complexity to solve OSCA grows exponentially with the number of sniffers, unless $P = NP$.

Also, we have an inapproximability result for OSCA.

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

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

III. THE DISTRIBUTED ALGORITHM FOR OSCA

We develop a distributed algorithm to solve OSCA, referred to as DA-OSCA. The basic structure of DA-OSCA is based on the Linear Program (LP) rounding technique, where we first solve the LP relaxation of OSCA and then round the

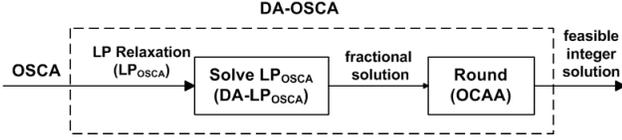


Fig. 1. Distributed Algorithm for OSCA (DA-OSCA).

(fractional) solution of the LP relaxation to an feasible integer solution to the original OSCA problem. Figure 1 shows an overview of how DA-OSCA yields an approximate solution to OSCA. DA-OSCA consists of two components: 1) the Distributed Algorithm to solve the LP relaxation of OSCA (DA-LP_{OSCA}); 2) Opportunistic Channel Assignment Algorithm (OCAAA) to perform distributed rounding of the fractional solution yielded by DA-LP_{OSCA}.

A. Distributed Algorithm for Solving LP relaxation of OSCA

LP relaxation of OSCA. We first formulate an integer linear program (ILP) of OSCA. We assign an indicator variable $x_n \in \{0, 1\}$ to each node $n \in N$, where $x_n = 1$ indicates that node n is covered by the given solution. We assign an indicator variable $y_{s,c} \in \{0, 1\}$ to a coverage-set $K_{s,c} \in \mathcal{K}$, and $y_{s,c} = 1$ indicates that sniffer s will be tuned to channel c . The ILP of OSCA, denoted by ILP_{OSCA}, is given by:

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

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

$$\sum_{c \in C} y_{s,c} \leq 1 \quad \forall s \in S, \quad (3)$$

$$0 \leq x_n, y_{s,c} \leq 1 \quad \forall n \in N, s \in S, c \in C, \quad (4)$$

$$x_n, y_{s,c} \in \{0, 1\} \quad \forall n \in N, s \in S, c \in C. \quad (5)$$

The objective function (1) together with the constraints (2) and (5) makes $x_n = 1$ if at least one coverage-set that includes the node n is chosen for a solution, and $x_n = 0$ otherwise. Eq. (3) is due to the group budget constraint.

Since ILP_{OSCA} cannot be solved in polynomial time, we relax the integer constraint (5) to obtain the LP relaxation of OSCA, i.e., Eqs. (1)–(4), denoted by LP_{OSCA}. In LP_{OSCA}, the variables x_l 's and y_{ij} 's can now take any value in $[0, 1]$, including fractional values.

Solving LP_{OSCA}. We use the *Proximal Optimization Algorithm* (POA) [13, Ch. 3.4.3] combined with a dual approach to solve LP_{OSCA}. POA introduces a set of auxiliary variables and adds quadratic terms to the objective function (1) of LP_{OSCA} to transform LP_{OSCA} into a quadratic program (QP) (as given in Eq. (6)), and then solves the QP by sequentially updating the values of the two kinds of variables, i.e. first the original variables and then the auxiliary variables. The rationale behind the transformation is to resolve a difficulty due to the linearity of the objective function (1) when we solve the dual problem of LP_{OSCA}. Specifically, the objective function (1) of LP_{OSCA} is linear, and hence it is not strictly concave. As a result, the

dual problem of LP_{OSCA} may not be differentiable at every point. This leads to a difficulty when we use the Gradient Projection Algorithm [13, Ch. 3.3.2] to solve the dual problem. However, such a difficulty will be resolved with the QP, since the objective function of the QP_{OSCA} is strictly concave due to the added quadratic terms and thus is differentiable.

We now apply POA to LP_{OSCA}. We introduce a set of auxiliary variables $\{x_n^{\text{aux}}, y_{s,c}^{\text{aux}} : n \in N, s \in S, c \in C\}$, and transform LP_{OSCA} into the following equivalent quadratic program, denoted by QP_{OSCA}:

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

subject to Eqs. (2)–(4).

Here, d is a positive constant. It can be shown that solving QP_{OSCA} is equivalent to solving LP_{OSCA} (refer to Appendix for the proof of this claim). For notational simplicity, we define $\vec{x} = (x_n : n \in N)$ and $\vec{y} = (y_{s,c} : s \in S, c \in C)$, and define \vec{x}^{aux} and \vec{y}^{aux} similarly as \vec{x} and \vec{y} . The POA to solve QP_{OSCA}, referred to as POA-QP_{OSCA}, proceeds as follows. At t -th iteration, $t = 1, 2, 3, \dots$, POA-QP_{OSCA} executes the following two steps:

- S1: Fixing $\vec{x}^{\text{aux}} = \vec{x}^{\text{aux}}(t)$ and $\vec{y}^{\text{aux}} = \vec{y}^{\text{aux}}(t)$, solve QP_{OSCA} with respect to \vec{x} and \vec{y} . Let the solution obtained be $\vec{x}(t), \vec{y}(t)$.
- S2: Let $\vec{x}^{\text{aux}}(t+1) = \vec{x}(t)$ and $\vec{y}^{\text{aux}}(t+1) = \vec{y}(t)$.

POA-QP_{OSCA} can start with any initial values, i.e. any $\vec{x}^{\text{aux}}(1)$ and $\vec{y}^{\text{aux}}(1)$. As the number t of iterations tends to infinity, a sequence of vectors generated by POA-QP_{OSCA} converges to the optimal solution of QP_{OSCA} [13, Ch. 3.4.3].

Note that, at Step S1 in each iteration of POA-QP_{OSCA}, we still have an optimization problem to be solved. We solve the optimization problem given at Step S1 by solving its dual problem instead. The reason why we solve the dual problem instead of the primal problem is that the dual problem has a simple form of constraints and is easily decomposable, and these features enable us to design a distributed algorithm to solve the problem.

We derive the dual problem of the optimization problem given by Step S1 of POA-QP_{OSCA}, i.e., the QP_{OSCA} with \vec{x}^{aux} and \vec{y}^{aux} being fixed. For notational simplicity, we let $\vec{z} = (\vec{x}, \vec{y})$ and $\vec{z}^{\text{aux}} = (\vec{x}^{\text{aux}}, \vec{y}^{\text{aux}})$. We define a set Z that contains all of (\vec{x}, \vec{y}) 's satisfying Eqs. (3) and (4). We define a set of Lagrange Multipliers $\vec{p} = (p_n : n \in N)$ for the $|N|$ constraints in Eq. (2). We define the Lagrangian function of the QP_{OSCA} with fixed \vec{x}^{aux} and \vec{y}^{aux} as

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

The dual problem is then given by

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

Since the dual objective function D in (8) is now differentiable due to the quadratic terms in Eq. (7), we can use the Gradient Projection Algorithm (GPA) (refer to [13, Ch. 3.3.2]) to solve the dual problem. The GPA to solve the dual problem has the following iterations: for $i = 0, 1, 2, \dots$,

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

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

Here, $\beta > 0$ is the step size, $[\cdot]_A^+$ 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 that maximizes $L(\vec{z}, \vec{p}(i); \vec{z}^{\text{aux}})$ for given $\vec{p}(i)$. To compute the iterations in Eq. (9), at each iteration, we need to solve the following maximization problem : for given $\vec{p}(i)$,

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

To solve Eq. (10), we rearrange the terms in Eq. (7) and rewrite Eq. (7) as the following:

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

Using Eq. (11), we can decompose the problem in Eq. (10) into the following sets of independent subproblems:

1) for each $n \in N$,

$$\begin{aligned} & \text{maximize} && -\frac{1}{2d} (x_n - x_n^{\text{aux}})^2 + (w_n - p_n(i)) x_n \\ & \text{subject to} && 0 \leq x_n \leq 1 \end{aligned} \quad (12)$$

2) for each $s \in S$,

$$\begin{aligned} & \text{maximize} && \sum_{c \in C} \left(-\frac{1}{2d} (y_{s,c} - y_{s,c}^{\text{aux}})^2 + y_{s,c} \sum_{n \in K_{s,c}} p_n(i) \right) \\ & \text{subject to} && \sum_{c \in C} y_{s,c} \leq 1 \text{ and } y_{s,c} \geq 0 \quad \forall c \in C. \end{aligned} \quad (13)$$

Note that each sub-problem can be solved independently at each node and at each sniffer, using purely local communication. By solving each subproblem independently, we can

Algorithm 1 DA-LP_{OSCA}

```

1: while TRUE do
2:   // Step 1 of POA-QPOSCA
3:   for  $i = 0$  to  $I \rightarrow \infty$  do
4:     Each node  $n$  and each sniffer  $s$  compute  $x_n(i)$  and  $\vec{y}_s(i)$  according to Eqs. (14) and (15), respectively. Then, 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-QPOSCA
10:  Each node  $n$  and each sniffer  $s$  set initial values of their variables for the next iteration as
      
$$x_n^{\text{aux}} \leftarrow x_n(I) \text{ and } p_n(0) \leftarrow p_n(I) \quad (\text{node } n)$$

      
$$\vec{y}_s^{\text{aux}} \leftarrow \vec{y}_s(I) \quad (\text{sniffer } s).$$

11: end while

```

obtain the solutions to Eqs. (12) and (13) as the following:

$$x_n^*(i) = [x_n^{\text{aux}} + d(w_n - p_n(i))]_{[0,1]}^+ \quad (14)$$

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

$$Y_s = \left\{ \vec{y}_s \triangleq (y_{s,c} : c \in C) : \sum_{c \in C} y_{s,c} \leq 1, y_{s,c} \geq 0 \quad \forall c \right\}. \quad (15)$$

Here, the projection $[\cdot]_{Y_s}^+$ in (15) can be easily done, e.g., with Alg. 6 in Appendix. Thus, we now have the solution to the dual problem (8). To solve the dual problem, we iteratively update the dual variables \vec{p} according to Eq. (9). Here, at each iteration, we need to compute $g_n(i)$, and this requires to solve the independent problems in Eqs. (12) and (13). To solve them, we update the primal variables \vec{x} and \vec{y} according to Eqs. (14) and (15).

Consequently, we finally have the solution to the Step S1 of POA-QP_{OSCA}. We obtain the solution by alternately updating the dual and the primal variables, according to Eq. (9) and Eqs. (14), (15), respectively. As the number i of iterations tends to infinity, a sequence of vectors given by Eq. (9) converges to the optimal solution of the dual problem [13, Proposition 3.4]. Once the optimal solution of the dual problem is obtained, we can find the optimal solution of the primal problem (i.e. the optimization problem given by Step S1 of POA-QP_{OSCA}) using (14) and (15) [14, Ch. 5.5.3].

To summarize, we present a formal description of the overall procedure to solve LP_{OSCA} in Alg. 1, which we refer to as the Distributed Algorithm for solving LP_{OSCA} (DA-LP_{OSCA}). Note that DA-LP_{OSCA} requires *only local communications* among

neighboring nodes. In many monitoring applications, it would be desirable that DA-LP_{OSCA} should be run by only sniffers since DA-LP_{OSCA} is for sniffers to determine their channels. In such cases, we can let one of neighboring sniffers of node n act as a proxy and take over the node n 's duty of updating values of the variables x_n , x_n^{aux} and p_n . Hence, each sniffer s needs to update values of its own variables \vec{y}_s , \vec{y}_s^{aux} , and also variables x_n 's, x_n^{aux} 's and p_n 's for some of its neighboring nodes. Since now sniffers update also the variables of nodes, each sniffer only needs to communicate with its neighboring sniffers to obtain the required values for the update of its variables.

DA-LP_{OSCA} with $I = 1$. The standard POA [13, Ch. 3.4.3], which is the DA-LP_{OSCA} when $I \rightarrow \infty$, requires a two-level convergence structure. That is, 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 because it increases the running time of DA-LP_{OSCA} and also incurs substantial communication overheads, due to a mechanism required to determine when to stop inner-level iterations. This intuition is that, as the number of inner-level iterations increases, the improvement of the solution quality at each iteration would decrease. Hence, such later iterations that give a small improvement would be wasteful, since solving the problem given by Step S1 is only an intermediate step to solve the ultimate problem. For these reasons, we fix the number of inner-level iterations of DA-LP_{OSCA} to 2 (i.e. $I = 1$), and find a good approximate solution.

We now show that, even with $I = 1$, DA-LP_{OSCA} can converge to the optimal solution. We let $\bar{z}^{\text{aux},t}$ and \bar{p}^t be the values of $\bar{z}^{\text{aux}}(I)$ and $\bar{p}(I)$, respectively, at the t -th outer-level iteration. Also, we let $\bar{z}^{\text{aux},*}$ and \bar{p}^* be the primal optimal solution and the dual optimal solution, respectively, of QP_{OSCA}. The following theorem¹ provides a sufficient condition of the step size β (to solve the dual problem Eq. (9)) for DA-LP_{OSCA} with $I = 1$ to converge.

Theorem 3: As $t \rightarrow \infty$, a sequence of vectors $(\bar{z}^{\text{aux},t}, \bar{p}^t)$ given by DA-LP_{OSCA} with $I = 1$ converges to $(\bar{z}^{\text{aux},*}, \bar{p}^*)$, provided that

$$\beta < \frac{1}{2dB_1B_2}, \text{ where}$$

$$B_1 = \max\{1, |K_{s,c}| : s \in S, c \in C\} + 1, \\ B_2 = \max\{|C|, M + 1\}, \text{ and } M = \max_{n \in N} \{|K_{s,c}| : n \in K_{s,c}\}.$$

The proof is given in Appendix. Here, the upper bound $\frac{1}{2dB_1B_2}$ can be obtained by computing the two pieces of information:

¹Our result in Theorem 3 can be viewed as a parallel version of the improved POA scheme [15], which has studied a cross-layer transmission scheduling problem in wireless networks. This work has previously used the idea of fixing the number of inner-level iterations. But, the results in [15] are based on the assumption that the coefficients in the constraints of the underlying LP problem must be non-negative. Hence, the results in [15] cannot be directly applied to our problem, i.e., LP_{OSCA} that have negative coefficients in the constraints.

the maximum number of node that can be covered by any sniffer operating on any channel, and the maximum number of neighboring sniffers that a normal node has.

B. Opportunistic Channel Assignment Algorithm

We develop a distributed rounding algorithm that determines the channel assignment of sniffers based on the optimal solution \vec{y}^* given by DA-LP_{OSCA}. We refer to this as the *Opportunistic Channel Assignment Algorithm* (OCAA). OCAA can be viewed as a distributed generalization of a centralized rounding scheme called PIPAGE [16]. PIPAGE guarantees that, for a given LP-relaxation solution that achieves a constant factor α of the optimal value of the LP relaxation, the integer solution yielded by PIPAGE always achieves at least $\alpha \cdot (1 - \frac{1}{e})$ of the optimal value of the original ILP. However, PIPAGE is not suitable for distributed solutions because PIPAGE rounds the LP-relaxation solution through a number of iterations and each iteration requires a global communication to evaluate the quality of the intermediate solution. On the other hand, our OCAA can achieve the same ratio $1 - \frac{1}{e}$ in a distributed manner that requires only local communications among neighboring sniffers. In this subsection, we first describe OCAA and then present the guarantee of OCAA.

We first introduce a metric called *coverage improvement* that guides each sniffer to make a good decision on selecting its channel. For a given set of values $\vec{y}_{N(s)}^* = \{y_{s',c}^* : s' \in N(s), c \in C\}$, where $N(s)$ denotes the set of neighboring sniffers of sniffer s , the *coverage improvement* of coverage-set $K_{s,c}$ is defined as

$$I(K_{s,c}; \vec{y}_{N(s)}^*) = \sum_{n \in K_{s,c}} w_n \left(\prod_{(s',c): s' \neq s, n \in K_{s',c}} (1 - y_{s',c}^*) \right). \quad (16)$$

Intuitively, by viewing $y_{s',c}^*$ as the probability that sniffer s' tunes its radio to channel c , we can interpret $I(K_{s,c}; \vec{y}_{N(s)}^*)$ as an expected coverage improvement, in terms of the total weight of the nodes in $K(s,c)$, that can be achieved by sniffer s tuning its ratio to channel c . Specifically, when $y_{s',c}^*$ is viewed as such a probability, $I(K_{s,c}; \vec{y}_{N(s)}^*)$ means the expected total weight of the *uncovered* nodes in $K(s,c)$, provided that all the neighboring sniffers of s (i.e., all s') do not tune their channels to c . In other words, $I(K_{s,c}; \vec{y}_{N(s)}^*)$ is the expected total weight improvement that sniffer s can achieve by tuning its radio to channel c . Note that sniffer s can compute its coverage improvements over all the channels by communicating only with its neighbors.

We formally present OCAA in Alg. 2. OCAA determines the channels of sniffers through several iterations, in the order according \mathcal{P} . In each iteration, the sniffers in P_i determine their channels in parallel such that each sniffer s selects the channel that achieves the maximum coverage improvement in terms of $I(K_{s,c}; \vec{y}_{N(s)}^*)$ for a fixed set of values $\vec{y}_{N(s)}^*$ for its neighbors (line 4). Thereafter, the sniffers that have determined their channels send the determination to their neighbors (line 5), so that, in the next iteration, some of the

Algorithm 2 Opportunistic Channel Assignment Algorithm

```

1: // Assume a partition  $\mathcal{P} = \{P_i\}$  of the set  $S$  of all sniffers
   such that no two sniffers in any  $P_i$  are neighbors.
2: for  $i = 1$  to  $|\mathcal{P}|$  do
3:   // All sniffers in  $P_i$  can choose their channels in parallel.
4:   Each sniffer  $s \in P_i$  tunes its radio to a channel  $c^* \in C$ 
   such that

```

$$I(K_{s,c^*}; \vec{y}_{N(s)}^*) = \max_{c \in C} I(K_{s,c}; \vec{y}_{N(s)}^*).$$

```

5:   After determining its channel, the sniffer  $s$  sends the
   determination to its neighboring sniffers.
6: end for

```

neighbors (in P_{i+1}) can use the determination to compute their coverage improvements. Here, the sequence \mathcal{P} can be determined a priori or through an ad hoc coordination among sniffers, e.g., employing one of existing scheduling algorithms at the Medium Access Control (MAC) layer.

Theorem 4: Given an solution to LP_{OSCA} that attains a constant factor α of the optimal value of LP_{OSCA} , OCAA guarantees to achieve at least $\alpha \cdot (1 - \frac{1}{e})$ ($\approx 0.632\alpha$) of the maximum monitoring coverage of OSCA.

The proof is given in Appendix. Here, the factor α comes from the approximate solution of LP_{OSCA} . However, note that we can make the approximate solution arbitrarily close to the optimal solution of LP_{OSCA} as we increase the number of outer-level iterations of $\text{DA-LP}_{\text{OSCA}}$. Hence, due to Theorems 3 and 4, we finally have the following theorem.

Theorem 5: DA-OSCA can always achieve at least $1 - \frac{1}{e}$ (≈ 0.632) of the maximum monitoring coverage of OSCA, regardless of the network topology and the channel assignment of nodes.

IV. ONLINE IMPLEMENTATION OF DA-OSCA

In this section, we present how to implement DA-OSCA to operate online so that DA-OSCA is agile and adapts incrementally to network changes, such as, changes to the channels assigned to nodes, changes in the usage of its channel by a node, and network topology changes due to mobility of nodes or arrivals/departures of sniffers. We present two operational modes of DA-OSCA—Mode-I and Mode-II, that are suitable for fast-varying and slow-varying networks, respectively. By developing the two operational modes, we enable DA-OSCA to operate in a more cost-effective manner for the two types of dynamic networks.

We first describe the procedure that sniffers need to perform, commonly for both operational modes, when they find arrivals/departures of their neighboring nodes/sniffers. Note that failures and recoveries of nodes/sniffers can be viewed as their departures and arrivals, respectively.

A. Basic information update

When sniffer s finds arrivals or departures of its neighboring nodes, it first updates its coverage-sets (i.e. \mathcal{K}_s). For the arrival

Algorithm 3 DA-OSCA in Mode-I

```

1: if  $t = k \cdot T_1, \forall k = 1, 2, \dots$  then
2:   Perform one outer-iteration of  $\text{DA-LP}_{\text{OSCA}}$  (i.e., lines 3–
   11 of Alg. 1)
3:   if  $t = k \cdot (lT_1), \forall k = 1, 2, \dots$  then
4:     Invoke OCAA
5:   end if
6: end if

```

of a new neighboring node n , the sniffer s that acts as a proxy for node n (for updating values of the node n 's variables) introduces a set of new variables for node n , i.e., x_n, x_n^{aux} and p_n , and sets their initial values as follows: $x_n = 1$ if node n is covered (by any of its neighboring sniffers), and otherwise $x_n = 0$; $x_n^{\text{aux}} = x_n$; $p_n = 0$. For the departure of its neighboring node n , the sniffer s removes the set of the variables for node n . When new sniffer s arrives, it first creates its coverage-sets and its variables, i.e., \vec{y}_s and \vec{y}_s^{aux} , 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 (according to Eq. (16)), and $y_{s,c} = 0$ for all $c \neq c^* \in C$; $\vec{y}_s^{\text{aux}} = \vec{y}_s$. When sniffer s leaves, one of its neighboring sniffers takes over the proxy duty that sniffer s had been doing.

B. Mode-I: DA-OSCA for fast-varying networks

In this mode, DA-OSCA operates *proactively* to adapt to frequent network changes. The rationale behind this proactive mode is that, when the network changes frequently, it is cost-effective to run DA-OSCA continuously, rather than running it on demand. This is because, as we will see in Mode-II, such a reactive operation of DA-OSCA will require global communications to evaluate the quality of the current monitoring coverage to determine when to start and also when to terminate. This process is costly.

The operation of DA-OSCA in Mode-I is presented in Alg 3. DA-OSCA executes one outer-level iteration of $\text{DA-LP}_{\text{OSCA}}$ every T_1 time (line 2), and invokes OCAA every lT_1 , i.e., every l outer-level iterations of $\text{DA-LP}_{\text{OSCA}}$ (line 4). Intuitively, DA-OSCA keeps updating the primal and the dual variables (using $\text{DA-LP}_{\text{OSCA}}$) and periodically change the channel assignment of sniffers based on the updated values of \vec{y} .

C. Mode-II: DA-OSCA for slow-varying networks

In this mode, DA-OSCA operates *on demand*, i.e., only when it needs to change the channel assignment of sniffers to improve the degraded monitoring coverage. For this reactive operational mode, DA-OSCA needs a mechanism to evaluate the quality of monitoring coverage to determine whether the invocation of DA-OSCA is needed, and also to check whether the iterations of $\text{DA-LP}_{\text{OSCA}}$ are sufficiently close to the optimal solution so that DA-OSCA should terminate $\text{DA-LP}_{\text{OSCA}}$ and round the solution with OCAA. Hence, in this subsection, we first develop a procedure to evaluate the

Algorithm 4 An efficient information-aggregation procedure to evaluate 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 before the root sniffer. At a level of the spanning tree, sniffer s computes:

$$\begin{aligned}
 C_s &= \sum_{s' \in \text{CS}(s)} C_{s'} + \sum_{n \in L(s)} w_n \cdot \min \left\{ 1, \sum_{(s,c):n \in K_{s,c}} y_{s,c} \right\} \\
 D_s &= \sum_{s' \in \text{CS}(s)} D_{s'} + \sum_{n \in K_{s,c^*}} p_n + \sum_{n \in L(s)} [w_n - p_n]^+,
 \end{aligned} \tag{17}$$

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

- 3: **Determination of solution quality.** The root sniffer computes C_{root} and D_{root} according to Eq. (17), and makes a decision of the termination of DA-LP_{OSCA} as follows: if $C_{\text{root}} \geq \gamma \cdot D_{\text{root}}$, then determines that the current channel assignment achieves the desired monitoring coverage. Thereafter, the root sniffer sends to 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.
-

quality of monitoring coverage, and then present how DA-OSCA employs the procedure to operate in the reactive mode.

We present an efficient information-aggregation procedure to evaluate the quality of monitoring coverage in Alg. 4. Basically, Alg. 4 estimates the gap between the current monitoring coverage and the maximum monitoring coverage, and then determines whether the estimate is above a desired level (that is specified by a pre-determined value of γ). Here, the gap is defined as the ratio of the current monitoring coverage to the maximum monitoring coverage. To estimate the gap, Alg. 4 computes the current monitoring coverage (i.e., C_{root}) and the dual objective function value (i.e., D_{root}) since it follows from the duality theory [14, Ch. 5.1.3] that any dual objective function is an upper bound on the primal optimal value, which is the optimal value of LP_{OSCA}, and thus is an upper bound on the maximum monitoring coverage. To compute them, Alg. 4 efficiently aggregates information through the spanning tree of sniffers (line 2), and then determines whether the current monitoring coverage is above the desired level by checking $C_{\text{root}} \geq \gamma \cdot D_{\text{root}}$ (line 3). Thus, this process does require collection of information in a hierarchical manner from all the sniffer nodes. Finally, the determination is distributed to all sniffers through the spanning tree. The proof of the correctness of Alg. 4 is given in Appendix.

We now describe how DA-OSCA operates on demand by

Algorithm 5 DA-OSCA in Mode-II

- 1: **if** $t = k \cdot T_2, \forall k = 1, 2, \dots$ **then**
 - 2: **if** $r_{\text{MC}} \leq \gamma_1$ (by invoking Alg. 4) **then**
 - 3: // i.e., when the ratio of the current monitoring coverage to the maximum possible monitoring coverage is below a desired level γ_1
 - 4: **while** $r_{\text{LP}} \leq \gamma_2$ (by invoking Alg. 4) **do**
 - 5: Perform N_o outer-iterations of DA-LP_{OSCA} (i.e., lines 3–11 of Alg. 1)
 - 6: **end while**
 - 7: Invoke OCAA
 - 8: **end if**
 - 9: **end if**
-

employing Alg. 4. We formally present the Mode-II of DA-OSCA in Alg. 5. In this mode, DA-OSCA evaluates the quality of the current monitoring coverage periodically, i.e., every T_2 time, by employing Alg. 4 (i.e., line 2 in Alg. 5). If the estimate (i.e., r_{MC}) of the gap between the current monitoring coverage and the maximum monitoring coverage is above a desired level, DA-OSCA terminates doing nothing (i.e., when the condition line 2 is not met). Otherwise, DA-OSCA starts to solve the new OSCA that has resulted from the network changes (lines 4–7). To solve the problem, DA-OSCA runs N_o outer-level iterations of DA-LP_{OSCA}. Here, N_o gives a trade-off between the cost due to checking the stopping criterion and the cost due to running more number of outer-level iterations of DA-LP_{OSCA} than required to reach the solution quality. Hence, N_o needs to be carefully chosen taking into account the convergence speed of DA-LP_{OSCA}. DA-OSCA checks whether the ratio r_{LP} of the solution of DA-LP_{OSCA} at the current iteration is sufficiently close to the optimal solution of LP_{OSCA} by employing Alg. 4 with a pre-specified precision of γ_2 (line 4). Once a near-optimal solution to LP_{OSCA} is obtained, DA-OSCA terminates DA-LP_{OSCA} and then rounds the solution of LP_{OSCA} with OCAA to obtain an integer solution. Then, DA-OSCA terminates.

V. NOTES

In OSCA, we assume that all of the nodes and the sniffers have only one radio. However, the case, where nodes and sniffers are equipped with multiple radios, can be easily mapped to this single-radio case by regarding radios of a node (or a sniffer) as different nodes (or sniffers) with a single radio. One might think that, the single-radio case, which is mapped from the multi-radio case, needs an additional constraint that ensures each sniffer to tune its radios to different channels. However, even without the additional constraint, our algorithm will automatically determine a set of distinct channels for each sniffer's radios. This is because tuning two radios of a sniffer to the same channel in the multi-radio case implies choosing two coverage-sets that contain the same nodes, and this always gives a lower coverage than choosing either of the two coverage-sets and any other coverage-set.

For OSCA, one could consider a simple randomized rounding scheme that views a channel assignment of a sniffer as a random experiment, where a random variable is assigned to each sniffer, and each random variable is realized to one of the available channels with a probability of its fractional value obtained by solving LP_{OSCA} (i.e. the LP relaxation of OSCA). It is easy to show (as in [6], [17]) that this randomized rounding scheme guarantees to achieve at least $1 - \frac{1}{e}$ (≈ 0.632) of the optimum of OSCA, in expectation. However, in order to achieve the expected guarantee of $1 - \frac{1}{e}$, the randomized rounding scheme requires sniffers to switch their channels a large number of times by repeatedly realizing their random variables with the same probability distribution. However, the delay of switching the radio channel is non-negligible². Hence, with this randomized rounding scheme, sniffers would waste their time switching channels. Thus, we use a deterministic rounding scheme, which does not require sniffers to switch their channels but can achieve the same approximation ratio $1 - \frac{1}{e}$ *deterministically*.

Theorem 3 suggests that the value of d (which is the coefficient of the quadratic term in the objective function (6) of QP_{OSCA}) should be small so that the step size β can be chosen to a large value, thus leading to a larger improvement at each inner-level iteration. On the other hand, a small value of d will cause the objective function (6) of QP_{OSCA} to be different from the objective function (1) of the original problem LP_{OSCA} , and hence require more outer-level iterations, thus potentially leading to slow convergence of DA- LP_{OSCA} . Therefore, the value of d should be tuned carefully.

VI. SIMULATION

We conduct simulations to demonstrate the efficacy of the two modes of DA-OSCA for two kinds of networks—random networks and scale-free networks. In random networks, nodes are randomly deployed with a uniform distribution. In scale-free networks, nodes are deployed such that the distribution $f(d)$ of nodes with degree d follows a power law in a form of d^{-r} . The performance of DA-OSCA largely depends on the network topology, and these two kinds of networks have a significant difference in their topologies. Also, their topologies are observed in many practical networks³.

We choose the settings of the network and the parameters of DA-OSCA as follows. There are 500 nodes of identical weight and 50 sniffers in the network. The number of available wireless channels is three (i.e., $|C| = 3$), same as the number of non-overlapping wireless channels in IEEE 802.11. For random networks, we randomly place nodes and sniffers on a 1×1 square area, and set the receiving range of sniffers to 0.15. For scale-free networks, the parameter r of the distribution $f(d) = O(d^{-r})$ is chosen as $2 < r < 3$. In

²Current estimate for switching delay between channels in the same frequency band with commodity IEEE 802.11 hardware is in the range of a few milliseconds [18] to a few hundred microseconds [19].

³Wireless networks where mobile users move randomly can be viewed as random networks, and many empirically observed networks, such as the world wide web and the Internet, have been found to be scale-free.

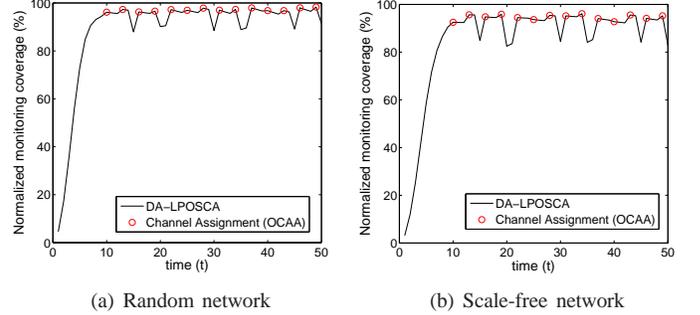


Fig. 2. Mode-I: DA-OSCA for fast-varying networks where the LP rounding executes continuously with updated coverage information.

scale-free networks, we pick nodes with highest degrees as sniffers. This is reasonable because thereby we can achieve a higher monitoring coverage than picking them randomly. The parameters of DA-OSCA are set as $S = 1$ (i.e., the number of inner-level iterations is 2), $d = 0.5$, and $\beta = 1/(B_1 B_2)$.

We conduct two experiments in each network. In one experiment, we evaluate the Mode-I of DA-OSCA in fast-varying networks, and in the other experiment, we evaluate the Mode-II of DA-OSCA in slow-varying networks. In all experiments, we demonstrate how monitoring coverage evolves as DA-OSCA adapts to the changes to the 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 slow-varying networks, respectively. Here, we one time unit as the time that DA-OSCA takes to run one outer-level iteration of DA- LP_{OSCA} . In Mode-I, we set the parameters as $T_1 = 1$ and $l = 3$. In Mode-II, we set the parameters as $T_2 = 30$, $\gamma_1 = 0.8$, $\gamma_2 = 0.8$, and $N_o = 1$. Here, we set the values of γ_1 and γ_2 taking into account that Alg. 4 underestimates the quality of monitoring coverage since its uses an upper bound on the maximum coverage. In all experiments, the results are the averages over 10 different network realizations.

Figure 2(a) and (b) show how the monitoring coverage evolves as DA-OSCA in Mode-I runs in a random networks and in a scale-free network, respectively. Here, the monitoring coverage is normalized by the optimal value of LP_{OSCA} , which is an upper bound on the maximum monitoring coverage. In this experiment, DA-OSCA adjusts the channel assignment of sniffers after 10 time units since the simulation begins. For both networks, we observe that the fractional monitoring coverage due to the solution of DA- LP_{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_{OSCA} quickly recovers the degraded fractional monitoring coverage, due to the changes of the channels assigned to nodes. Within only a few time units, the new channel assignment of sniffers by OCAA attains a high monitoring coverage, maintained

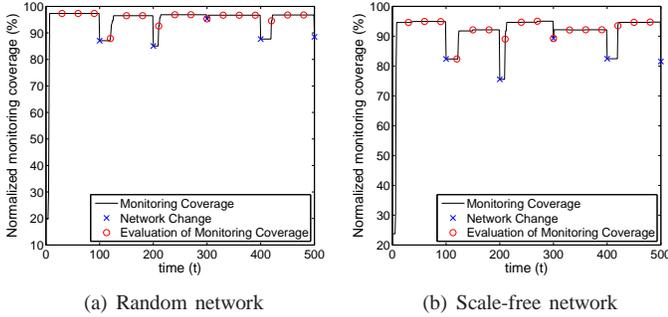


Fig. 3. Mode-II: DA-OSCA for slow-varying networks where the algorithm is executed on demand when a change is detected in the network.

above 95% of the maximum coverage. A notable difference between these results (also observed in Fig. 3(a), (b)) is that, in random networks, the channel changes of nodes incur less degradation of the monitoring coverage than in scale-free networks, and DA-OSCA achieves a higher monitoring coverage in random networks. This is, possibly, because in random networks sniffers are uniformly distributed and this makes sniffers have a better topological coverage than in scale-free networks.

Figure 3(a) and (b) demonstrate the on-demand operation of DA-OSCA in Mode-II for slow-varying networks. In both figures, we see observe large intervals of time where the monitoring coverage is flat. This means that, through Alg. 4, DA-OSCA determined that the monitoring coverage meets the desired level, and then terminates without any processing, thereby saving unnecessary cost. We notice that when the network changes, the monitoring coverage suffers (note the dips) but quickly recovers (always within 20 time units) as OCAA is executed on demand. Also, we observe that the improved monitoring coverage after the execution of DA-OSCA is higher than required (recall that $\gamma_2 = 0.8$). This can be explained by the following two facts. The first is that OCAA often improves the fractional solution while rounding it, which can be observed from Fig. 2(a) and (b). The second is that since Alg. 4 underestimates the quality of monitoring coverage, DA-OSCA may run the outer-iterations of DA-LP_{OSCA} more than required.

Both experiments show that DA-OSCA is able to adapt to different kinds of networks, fast-varying and slow-varying, and is able to operate incrementally with respect to network changes. By setting the values of γ , the system owner can control how close she wants the normalized monitoring coverage to get to the value of one.

VII. CONCLUSION

In this paper, we presented a distributed online algorithm for the optimal channel assignment problem for passive monitoring in multi-channel wireless networks. Our algorithm preserves the approximation ratio $1 - \frac{1}{e}$ that the existing centralized algorithms have previously attained, while providing a distributed solution that is amenable to online implementation.

We present two operational modes of our algorithm for cost-effective operation in two types of networks that have different rates of network changes. Simulation results demonstrate the effectiveness of the two modes of our algorithm. Our future work is on how to make our distributed algorithm execute asynchronously. Further, we are studying the security monitoring problem where a node needs to be covered by multiple sniffers for reliable monitoring, due to imperfect sniffers.

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] P. Kyasanur, J. So, C. Chereddi, and N. H. Vaidya, “Multi-Channel Mesh Networks: Challenges and Protocols,” *IEEE Wireless Communications*, April 2006.
- [3] A. Raniwala and T. Chiueh, “Architecture and Algorithms for an IEEE 802.11-Based Multi-Channel Wireless Mesh Network,” in *Proc. of IEEE INFOCOM, Miami, FL, USA*, March 2005.
- [4] 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.
- [5] 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, Barcelona, Spain*, August, Ed., 2009.
- [6] D.-H. Shin and S. Bagchi, “Optimal Monitoring in Multi-Channel Multi-Radio Wireless Mesh Networks,” in *Proc. of ACM MobiHoc*, 2009.
- [7] 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.
- [8] P. Arora, C. Szepesvari, and R. Zheng, “Sequential Learning for Optimal Monitoring of Multi-channel Wireless Networks,” in *Proc. of IEEE INFOCOM*, 2011.
- [9] D. Subhadhrabandhu, 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.
- [10] —, “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.
- [11] —, “A Statistical Framework for Intrusion Detection in Ad Hoc Networks,” in *Proc. of the 25th IEEE International Conference on Computer Communications (INFOCOM’06), Barcelona Spain*, April 2006.
- [12] P. Kyasanur and N. H. Vaidya, “Routing and Link-layer Protocols for Multi-Channel Multi-Interface Ad Hoc Wireless Networks,” *SIGMOBILE Mobile Computing and Communications Review*, vol. 10, no. 1, pp. 31–43, January 2006.
- [13] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, New Jersey, 1989.
- [14] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [15] X. Lin and N. B. Shroff, “Utility Maximization for Communication Networks with Multi-path Routing,” *IEEE Transactions on Automatic Control*, 2006.
- [16] A. Ageev and M. Sviridenko, “A New Method of Constructing Algorithms with Proven Performance Guarantee,” *Jour. of Combinatorial Optimization*, 2004.
- [17] A. Srinivasan, “Distributions on Level-Sets with Applications to Approximation Algorithms,” in *Proc. of IEEE FOCS*, 2001.
- [18] R. Chandra, P. Bahl, and P. Bahl, “MultiNet: Connecting to Multiple IEEE 802.11 Networks Using a Single Wireless Card,” in *Proc. of IEEE INFOCOM*, 2004.
- [19] “Maxim 2.4 GHz 802.11b Zero-IF Transceivers,” in <http://pdfserv.maxim-ic.com/en/ds/MAX2820-MAX2821.pdf>.

APPENDIX

Proof of Claim in Section III-A: We show the claim in Section III-A that solving QP_{OSCA} is equivalent to solving LP_{OSCA}. Let $\{x_n^*, y_{s,c}^*, x_n^{\text{aux},*}, y_{s,c}^{\text{aux},*}\}$ be the optimal solution of

QP_{OSCA}. Note that all of the quadratic terms in the objective function (6) of QP_{OSCA} are non-positive, and also that there is no constraint on the variables $x_n^{\text{aux},*}$'s and $y_{s,c}^{\text{aux},*}$'s. Hence, in order to maximize the objective function (6), it must be true that $x_n^{\text{aux},*} = x_n^*$ and $y_{s,c}^{\text{aux},*} = y_{s,c}^*$. This means that $\{x_n^*, y_{s,c}^*\}$ maximizes $\sum_{n \in N} w_n x_n$ subject to Eqs. (2)–(4) and thus is an optimal solution to LP_{OSCA}. Therefore, we can find an optimal solution to LP_{OSCA} by solving QP_{OSCA}. Thus, the claim is true. ■

Derivation of Alg. 6:

Let \vec{v}^{+v} be the projection of \vec{v} to V . With definition of projection, i.e., $\vec{v}^{+v} = \operatorname{argmin}_{\vec{x} \in V} d(\vec{v}, \vec{x})$ where $d(\vec{v}, \vec{x})$ denotes the Euclidean distance between \vec{v} and \vec{x} , it is easy to verify that if $v_j \leq 0$, then $v_j^{+v} = 0$. In order to obtain v_j^{+v} for $v_j > 0$, we redefine \vec{v} by removing the negative and zero components from \vec{v} . We assume that the dimension of the redefined vector \vec{v} is $d \leq c$. We also redefine $V = \{\vec{x} = (x_1, \dots, x_d) : x_j \geq 0 \text{ for all } j \in \{1, \dots, d\} \text{ and } \sum_{j=1}^d x_j \leq 1\}$. The problem then becomes to find the projection of the redefined vector $\vec{v} > 0$ to V .

Obviously, if $\vec{v} \in V$, $\vec{v}^{+v} = \vec{v}$. Hence, we only need to consider the case when $\vec{v} \notin V$. In this case, \vec{v} must be included in the set $U = \{\vec{x} : \sum_{j=1}^d x_j > 1 \text{ and } x_j > 0 \text{ for all } j \in \{1, \dots, d\}\}$. We define a bounded hyperplane $F = \{\vec{x} : \sum_{j=1}^d x_j = 1 \text{ and } x_j \geq 0 \text{ for all } j \in \{1, \dots, d\}\}$, and define $H = \{\vec{x} : \sum_{j=1}^d x_j = 1\}$ to be the hyperplane that includes F . Due to the following lemma, we only need to find $[\vec{v}^{\perp H}]_F^+$ in order to obtain \vec{v}^{+v} .

Lemma 1: For any $\vec{v} \in U$, $\vec{v}^{+v} = [\vec{v}^{\perp H}]_F^+$, where $\vec{v}^{\perp H}$ denotes the perpendicular foot of \vec{v} onto the hyperplane H .

Proof: To prove the lemma, we first show that \vec{v}^{+v} is a point on the bounded hyperplane F . To show this claim, we only need to show that the line segment that connects any $\vec{v} \in U$ and any $\vec{x} \in V$, denoted by $\overline{v\vec{x}}$, intersects with F . It is because if there exists a point at which $\overline{v\vec{x}}$ intersects with F , denoted by \vec{y} , the distance between \vec{v} and \vec{y} would be smaller than or equal to the distance between \vec{v} and \vec{x} , which implies that $\vec{v}^{+v} \in F$. In order to show the claim, we consider the line that passes through the points \vec{v} and \vec{x} , denoted by $\overleftrightarrow{v\vec{x}}$. The line $\overleftrightarrow{v\vec{x}}$ is a set of points $\{\vec{x} + t(\vec{v} - \vec{x}) : t \text{ is a real number}\}$. This line intersects with the hyperplane H at the point $\vec{p} = \vec{x} + t(\vec{v} - \vec{x})$, where $t = \frac{1 - \sum_{j=1}^d x_j}{\sum_{j=1}^d v_j - \sum_{j=1}^d x_j}$. Since $\vec{v} \in U$ and $\vec{x} \in V$, it is true that $0 \leq t < 1$. This implies that $\vec{p} \in \overline{v\vec{x}}$ and also that $\vec{p} > 0$. Also, due to the facts that $\vec{p} \in H$ and that $\vec{p} > 0$, it follows that $\vec{p} \in F$. Hence, $\overline{v\vec{x}}$ intersects with F at the point \vec{p} , and thus the claim is true, i.e., $\vec{v}^{+v} \in F$. Then, $\vec{v}^{+v} = \operatorname{argmin}_{\vec{x} \in F} d(\vec{v}, \vec{x})$. By Pythagorean theorem, it follows that $d(\vec{v}, \vec{x})^2 = d(\vec{v}, \vec{v}^{\perp H})^2 + d(\vec{v}^{\perp H}, \vec{x})^2$ for any $\vec{x} \in F$. Here, $d(\vec{v}, \vec{v}^{\perp H})$ is a constant. Hence, $\vec{v}^{+v} = \operatorname{argmin}_{\vec{x} \in F} d(\vec{v}^{\perp H}, \vec{x})$, i.e., $\vec{v}^{+v} = [\vec{v}^{\perp H}]_F^+$. ■

We find $[\vec{v}^{\perp H}]_F^+$ in a recursive manner. Let $\vec{v}^{+, (0)} = [\vec{v}^{\perp H}]_F^+$. A simple calculation gives $\vec{v}^{\perp H} = (v_1 + t, \dots, v_d + t)$ where $t = \frac{1}{d}(1 - \sum_{j=1}^d v_j)$. If $\vec{v}^{\perp H} \in \mathcal{F}$, $\vec{v}^{+, (0)} = \vec{v}^{\perp H}$. Otherwise, i.e., if $\vec{v}^{\perp H} \notin \mathcal{F}$, at least one component of

Algorithm 6 Projection Algorithm

```

1: // Algorithm projects  $\vec{v}$  to  $V = \{(x_1, \dots, x_c) : x_j \geq 0 \text{ for all } j \in \{1, \dots, c\} \text{ and } \sum_{j=1}^c x_j \leq 1\}$ .
2:  $J \leftarrow \{1, \dots, c\}$ 
3: while (1) do
4:   for  $j \leftarrow 1$  to  $|J|$  do
5:     if  $v_{J_j} \leq 0$  (where  $J_j$  denotes the  $j$ -th element of  $J$ ) then
6:        $v_{J_j} \leftarrow 0$ 
7:        $J \leftarrow J \setminus \{J_j\}$ 
8:     end if
9:   end for
10:  // Here, it is invariant that  $v_j > 0$  for all  $j \in J$ , and also that  $v_j = 0$  for all  $j \notin J$ .
11:  if  $|J| = 0$  or  $\sum_{j=1}^{|J|} v_{J_j} \leq 1$  then
12:    Terminate the algorithm
13:  else
14:    for  $j \leftarrow 1$  to  $|J|$  do
15:       $v_{J_j} \leftarrow v_{J_j} + \frac{1}{|J|} \left(1 - \sum_{j=1}^{|J|} v_{J_j}\right)$ 
16:    end for
17:    // Here, it is invariant that  $\sum_{i=1}^c v_i = 1$ .
18:  end if
19: end while
20: return  $\vec{v}$ 

```

$\vec{v}^{\perp H}$ must have a negative value since $\vec{v}^{\perp H} \in H$. It is easy to verify that the components of $\vec{v}^{+, (0)}$ corresponding to those of $\vec{v}^{\perp H}$ that have a negative value or zero must be zero. Without loss of generality, we assume that the positive components of $\vec{v}^{\perp H}$ are $v_1^{\perp H}, \dots, v_e^{\perp H}$ where $e \leq d - 1$. Since $\sum_{j=1}^d v_j^{\perp H} = 1$ and $\vec{v}^{\perp H}$ has at least one negative component, it follows that $\sum_{j=1}^e v_j^{\perp H} > 1$. Let $\vec{v}^{(1)} = (v_1^{\perp H}, \dots, v_e^{\perp H})$ and $U^{(1)} = \{(x_1, \dots, x_e) : \sum_{j=1}^e x_j > 1 \text{ and } x_j > 0 \text{ for all } j \in \{1, \dots, e\}\}$, then $\vec{v}^{(1)} \in U^{(1)}$. Define $F^{(1)} = \{(x_1, \dots, x_e) : \sum_{j=1}^e x_j = 1 \text{ and } x_j > 0 \text{ for all } j \in \{1, \dots, e\}\}$ and $H^{(1)} = \{(x_1, \dots, x_e) : \sum_{j=1}^e x_j = 1\}$. We then have $(v_1^{+, (0)}, \dots, v_e^{+, (0)}) = [\vec{v}^{(1)}]_{F^{(1)}}^+$ since $v_{e+1}^{+, (0)}, \dots, v_d^{+, (0)}$ are all zeros. Using Pythagorean theorem, we get $(v_1^{+, (0)}, \dots, v_e^{+, (0)}) = [\vec{v}^{(1)\perp H^{(1)}}]_{F^{(1)}}^+$, where $\vec{v}^{(1)\perp H^{(1)}}$ denotes the perpendicular foot of $\vec{v}^{(1)}$ onto the hyperplane $H^{(1)}$. The problem of finding $[\vec{v}^{\perp H}]_F^+$ then becomes to find $[\vec{v}^{(1)\perp H^{(1)}}]_{F^{(1)}}^+$. Note that both the problems differ only in the dimension of the vector. Also, the dimension of the vector in the former problem is at least one less than that in the latter problem. Hence, in order to find $[\vec{v}^{(1)\perp H^{(1)}}]_{F^{(1)}}^+$, we can repeat the process that we have done to find $[\vec{v}^{\perp H}]_F^+$. At the n -th iteration of this process, we would be able to obtain $[\vec{v}^{(n-1)\perp H^{(n-1)}}]_{F^{(n-1)}}^+$, equivalently $[\vec{v}^{\perp H}]_F^+$, or reduce the dimension of the vector by at least one. Since we start with the dimension $d \leq c$, the number of these iterations to obtain $[\vec{v}^{\perp H}]_F^+$ is at most c .

Alg. 6 implements this procedure to obtain the projection $[\vec{v}]_V^+$.

Proof of Theorem 3:

To show the theorem, we use the proof of Proposition 4 in [15]. For this, we first formulate the constraints (2)–(4) of QP-MC into the matrix form: $A\vec{z} \leq \vec{0}$, $\vec{z} \in Z$, where the matrix A is defined as

$$A = \left(\begin{array}{c|c} I_{|N|} & A_{\text{sub1}} \\ \hline O_{|S|,|N|} & A_{\text{sub2}} \end{array} \right) \in \mathbb{R}^{(|N|+|S|) \times (|N|+|S| \cdot |C|)},$$

where

$$A_{\text{sub1}} = \begin{pmatrix} -\mathbf{1}_{K_{S_1, C_1}}(N_1) & \cdots & -\mathbf{1}_{K_{S_{|S|}, C_{|C|}}}(N_1) \\ \vdots & \ddots & \vdots \\ -\mathbf{1}_{K_{S_1, C_1}}(N_{|N|}) & \cdots & -\mathbf{1}_{K_{S_{|S|}, C_{|C|}}}(N_{|N|}) \end{pmatrix} \in \mathbb{R}^{|N| \times (|S| \cdot |C|)},$$

$$A_{\text{sub2}} = \begin{pmatrix} 1 & \cdots & 1 & \cdots & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots & & \\ 0 & \cdots & 0 & \cdots & 1 & \cdots & 1 \end{pmatrix} \in \mathbb{R}^{|S| \times (|S| \cdot |C|)}.$$

Here, $I_{|N|}$ is $|N| \times |N|$ identity matrix, $O_{|S|,|N|}$ is $|S| \times |N|$ zero matrix, S_i denotes the i -th element of the set S , and $\mathbf{1}_S(s)$ is an indicator function defined as: $\mathbf{1}_S(s) = 1$ if $s \in S$; otherwise, $\mathbf{1}_S(s) = 0$.

Using the proof of Proposition 4 in [15], it can be shown that a sufficient condition for DA-LP_{OSCA} to converge is that $\frac{1}{\beta} I_{|N|+|S|} - 2dAA^T$ must be positive definite. The matrix $\frac{1}{\beta} I_{|N|+|S|} - 2dAA^T$ is positive definite if and only if for any non-zero vector \vec{s} ,

$$\vec{s}^T \left(\frac{1}{\beta} I_{|N|+|S|} - 2dAA^T \right) \vec{s} > 0$$

$$\iff \frac{1}{\beta} \sum_{i=1}^{|N|+|S|} s_i^2 > 2d (A^T \vec{s})^2. \quad (18)$$

It follows that

$$(A^T \vec{s})^2 = \sum_{j=1}^{|N|+|S| \cdot |C|} \left(\sum_{i=1}^{|N|+|S|} A_{i,j} s_i \right)^2$$

$$\leq \sum_{j=1}^{|N|+|S| \cdot |C|} \left(\sum_{i=1}^{|N|+|S|} |A_{i,j}| \right) \left(\sum_{i=1}^{|N|+|S|} |A_{i,j}| s_i^2 \right)$$

(by Cauchy-Schwartz inequality)

$$\leq \max_{\forall j} \left\{ \sum_{i=1}^{|N|+|S|} |A_{i,j}| \right\} \sum_{i=1}^{|N|+|S|} s_i^2 \sum_{j=1}^{|N|+|S| \cdot |C|} |A_{i,j}|$$

$$\leq \max_{\forall j} \left\{ \sum_{i=1}^{|N|+|S|} |A_{i,j}| \right\} \cdot \max_{\forall i} \left\{ \sum_{j=1}^{|N|+|S| \cdot |C|} |A_{i,j}| \right\}$$

$$\times \sum_{i=1}^{|N|+|S|} s_i^2. \quad (19)$$

Hence, $\frac{1}{\beta} I_{|N|+|S|} - 2dAA^T$ is positive definite if the following holds:

$$\beta < \frac{1}{2dB_1B_2},$$

where

$$B_1 = \max \left\{ \sum_{i=1}^{|N|+|S|} |A_{i,j}| : j \in [|N| + |S| \cdot |C|] \right\},$$

$$B_2 = \max \left\{ \sum_{j=1}^{|N|+|S| \cdot |C|} |A_{i,j}| : i \in [|N| + |S|] \right\},$$

where $[n]$ denotes an index set $\{1, \dots, n\}$. It follows that

$$B_1 = \max \left\{ 1, 1 + \sum_{l=1}^{|N|} \mathbf{1}_{K_{S_i, C_j}}(N_l) : i \in [|S|], j \in [|C|] \right\}$$

$$= \max \{1, |K_{s,c}| + 1 : s \in S, c \in C\},$$

and also that

$$B_2 = \max \left\{ |C|, 1 + \sum_{i=1}^{|S|} \sum_{j=1}^{|C|} \mathbf{1}_{K_{S_i, C_j}}(N_l) : l \in [|N|] \right\}$$

$$= \max \{|C|, M + 1\},$$

where $M = \max_{n \in N} |\{K_{s,c} : n \in K_{s,c}\}|$. Thus, the theorem follows. \blacksquare

Proof of Theorem 4: To prove the theorem, we show that OCAA is a distributed generalization of PIPAGE [16] that achieves the guarantee in the theorem in a centralized manner. For this, we first explain how PIPAGE solves OSCA. The PIPAGE applied to solve OSCA rounds a (fractional) solution of LP_{OSCA} to a feasible integer solution to ILP_{OSCA} in an iterative manner. Since each sniffer can assign only one channel to its radio, each sniffer has more than two non-integer values if it has non-integer values. At each iteration, PIPAGE adjusts two non-integer values of a sniffer such that at least one of them becomes an integer of 0 or 1, and the sum of them are preserved. Hence, when a sniffer has only two non-integer values, both of them will become an integer value of 0 or 1 after the adjustment by PIPAGE. At each iteration, PIPAGE adjusts two non-integer values of a sniffer as follows. Let $0 < y_{s,c_1}, y_{s,c_2} < 1$ be the two non-integer values of a sniffer to be adjusted at an iteration, and define $\epsilon_1 = \min\{y_{s,c_1}, 1 - y_{s,c_2}\}$ and $\epsilon_2 = \min\{1 - y_{s,c_1}, y_{s,c_2}\}$. At the iteration, PIPAGE adjusts the fractional solution \vec{y} including y_{s,c_1} and y_{s,c_2} to a new solution of either $\vec{y}^{(1)}$ or $\vec{y}^{(2)}$, which have the same values for all components except ones whose indices are (s, c_1) and (s, c_2) . In $\vec{y}^{(1)}$, the two components are $y_{s,c_1}^{(1)} = y_{s,c_1} - \epsilon_1$ and $y_{s,c_2}^{(1)} = y_{s,c_2} + \epsilon_1$, and in $\vec{y}^{(2)}$, they are $y_{s,c_1}^{(2)} = y_{s,c_1} + \epsilon_2$ and $y_{s,c_2}^{(2)} = y_{s,c_2} - \epsilon_2$ in $\vec{y}^{(2)}$. PIPAGE adjusts \vec{y} to $\vec{y}^{(1)}$ if $F(\vec{y}^{(1)}) \geq F(\vec{y}^{(2)})$, where $F(\vec{y}) = \sum_{n \in N} w_n \left(1 - \prod_{(s,c): n \in K_{s,c}} (1 - y_{s,c}) \right)$. Otherwise, PIPAGE adjusts \vec{y} to $\vec{y}^{(2)}$.

We now show OCAA accomplishes the procedure that the PIPAGE applied to solve OSCA performs. To show this, we first derive an efficient way of evaluating the criterion $F(\vec{y}^{(1)}) \geq F(\vec{y}^{(2)})$ that PIPAGE uses to adjust the fractional solution at each iteration. Since $y_{s,c_1} + y_{s,c_2} \leq 1$ due to the

group budget constraint, we have $\epsilon_1 = y_{s,c_1}$ and $\epsilon_2 = y_{s,c_2}$, and consequently we have

$$\begin{aligned} y_{s,c_1}^{(1)} &= 0, & y_{s,c_2}^{(1)} &= y_{s,c_1} + y_{s,c_2}, \\ y_{s,c_1}^{(2)} &= y_{s,c_1} + y_{s,c_2}, & y_{s,c_2}^{(2)} &= 0. \end{aligned}$$

It follows that

$$\begin{aligned} F(\vec{y}) &= \sum_{n \in N} w_n \left(1 - \prod_{(s,c):n \in K_{s,c}} (1 - y_{s,c}) \right) \\ &= \sum_{n \in N} w_n - \sum_{n \in N} w_n \left(\prod_{(s,c):n \in K_{s,c}} (1 - y_{s,c}) \right), \end{aligned}$$

and also that

$$\begin{aligned} &\sum_{n \in N} w_n \left(\prod_{(s,c):n \in K_{s,c}} (1 - y_{s,c}) \right) \\ &= \sum_{n \in K_{s,c_1}} w_n \left(\prod_{s' \neq s: n \in K_{s',c_1}} (1 - y_{s',c_1}) \right) (1 - y_{s,c_1}) \\ &+ \sum_{n \in K_{s,c_2}} w_n \left(\prod_{s' \neq s: n \in K_{s',c_2}} (1 - y_{s',c_2}) \right) (1 - y_{s,c_2}) \\ &+ \sum_{n \in N: n \notin K_{s,c_1}, n \notin K_{s,c_2}} w_n \left(\prod_{(s,c):n \in K_{s,c}} (1 - y_{s,c}) \right). \end{aligned}$$

Since $y_{s',c}^{(1)} = y_{s',c}^{(2)} = y_{s',c}$ for all $(s',c) \neq (s,c_1), (s,c_2)$ and $(y_{s,c_1}^{(1)} - y_{s,c_1}^{(2)}) = -(y_{s,c_2}^{(1)} - y_{s,c_2}^{(2)})$, it follows that

$$\begin{aligned} &F(\vec{y}^{(1)}) - F(\vec{y}^{(2)}) \\ &= \sum_{n \in K_{s,c_1}} w_n \left(\prod_{s' \neq s: n \in K_{s',c_1}} (1 - y_{s',c_1}) \right) \times (y_{s,c_1}^{(1)} - y_{s,c_1}^{(2)}) \\ &+ \sum_{n \in K_{s,c_2}} w_n \left(\prod_{s' \neq s: n \in K_{s',c_2}} (1 - y_{s',c_2}) \right) \times (y_{s,c_2}^{(1)} - y_{s,c_2}^{(2)}) \\ &= (I(K_{s,c_1}, \vec{y}_{N(s)}) - I(K_{s,c_2}, \vec{y}_{N(s)})) \times (y_{s,c_1}^{(1)} - y_{s,c_1}^{(2)}). \end{aligned}$$

Hence, since $y_{s,c_1}^{(1)} < y_{s,c_1}^{(2)}$, $F(\vec{y}^{(1)}) \geq F(\vec{y}^{(2)})$ if $I(K_{s,c_1}, \vec{y}_{N(s)}) \leq I(K_{s,c_2}, \vec{y}_{N(s)})$. This means that PIPAGE adjusts \vec{y} to $\vec{y}^{(1)}$ if $I(K_{s,c_1}, \vec{y}_{N(s)}) \leq I(K_{s,c_2}, \vec{y}_{N(s)})$. Otherwise, PIPAGE adjusts \vec{y} to $\vec{y}^{(2)}$.

Recall that when PIPAGE rounds non-integer values of the variables $\vec{y}_s = (y_{s,c} : c \in C)$ of sniffer s through multiple iterations, the values that are not in \vec{y}_s , i.e., $\tilde{y}_{s',c}$'s for all (s',c) such that $s' \neq s$, will remain the same. Hence, while the non-integer values of \vec{y}_s are rounded, the values of $I(K_{s,c}, \vec{y}_{N(s)})$'s for all $c \in C$ will remain the same. Therefore, after the multiple iterations to round the non-integer values of \vec{y}_s , all of the non-integer values except one that has the maximum coverage improvement among all non-integer values, say y_{s,c^*} , will be rounded to 0, and y_{s,c^*} will be adjusted to the sum

of all the non-integer values, which is equal to 1. This is the rounding procedure that OCAA performs. Thus, the theorem follows. \blacksquare

Proof of the correctness of Alg. 4: To show the correctness of Alg. 4, we use the duality theory [14, Ch. 5.1.3], which states that, for any maximization problem, the maximum of the given primal problem is upper bounded by the dual objective value of any feasible dual solution. To derive the dual problem of LP_{OSCA}, we define the Lagrangian function of LP_{OSCA} as

$$L_{LP}(\vec{z}, \vec{p}) = \sum_{n \in N} w_n x_n + \sum_{n \in N} p_n \left(\sum_{(s,c):n \in K_{s,c}} y_{s,c} - x_n \right). \quad (20)$$

The dual problem of LP_{OSCA} is then given as

$$\text{minimize } D_{LP}(\vec{p}) \triangleq \max_{\vec{z} \in Z} L_{LP}(\vec{z}, \vec{p}), \quad (21)$$

where Z is the set that contains all of (\vec{x}, \vec{y}) 's satisfying Eqs. (3) and (4). Let $F_{LP}(\vec{z}) = \sum_{n \in N} w_n x_n$, and $\tilde{\vec{z}}, \tilde{\vec{p}}$ be any feasible primal and dual solutions, respectively. Due to the duality theory [14, Ch. 5.1.3], it follows that for $0 < \gamma < 1$,

$$F_{LP}(\tilde{\vec{z}}) \geq \gamma \cdot D_{LP}(\tilde{\vec{p}}) \implies F_{LP}(\tilde{\vec{z}}) \geq \gamma \cdot F_{LP}^*, \quad (22)$$

where F_{LP}^* denotes the maximum of LP_{OSCA}.

We show the correctness of Alg. 4 using Eq. (22). For a given channel assignment of sniffers, which we denote by an integer vector \vec{y}^{int} , the monitoring coverage due to \vec{y}^{int} is given as $\sum_{n \in N} w_n x_n^{\text{int}}$, where $x_n^{\text{int}} = \min \left\{ 1, \sum_{(s,c):n \in K_{s,c}} y_{s,c}^{\text{int}} \right\}$, which is equal to C_{root} in Alg. 4. It is easy to see that $\vec{z}^{\text{int}} = (\vec{x}^{\text{int}}, \vec{y}^{\text{int}})$ is a feasible solution to LP_{OSCA}. We next compute $D_{LP}(\vec{p})$ for any given $\vec{p} \geq 0$. We rewrite Eq. (20) as

$$L_{LP}(\vec{z}, \vec{p}) = \sum_{n \in N} (w_n - \tilde{p}_n) x_n + \sum_{s \in S} \sum_{c \in C} \left(\sum_{n \in K_{s,c}} \tilde{p}_n \right) y_{s,c}. \quad (23)$$

For the given \vec{p} , we can obtain $\vec{z}^* \in Z$ that maximizes $L_{LP}(\vec{z}, \vec{p})$ subject to $\vec{z} \in Z$ as

$$\begin{aligned} x_n^* &= \begin{cases} 1 & \text{if } w_n \geq \tilde{p}_n, \\ 0 & \text{otherwise,} \end{cases} \\ y_{s,c}^* &= \begin{cases} 1 & \text{for } c^* \in \operatorname{argmax}_{c \in C} \sum_{n \in K_{s,c}} \tilde{p}_n, \\ 0 & \text{for all } c \neq c^*. \end{cases} \end{aligned} \quad (24)$$

Using Eqs. (21), (23), and (24), we can obtain $D_{LP}(\vec{p})$ for the given \vec{p} as

$$D_{LP}(\vec{p}) = \sum_{n \in N} [w_n - \tilde{p}_n]^+ + \sum_{s \in S} \sum_{n \in K_{s,c^*}} \tilde{p}_n,$$

where $c^* \in \operatorname{argmax}_{c \in C} \sum_{n \in K_{s,c}} \tilde{p}_n$. Hence, $D_{LP}(\vec{p})$ is equal to D_{root} in Alg. 4. Therefore, due to Eq. (22), if $C_{\text{root}} \geq \gamma \cdot D_{\text{root}}$, then $C_{\text{root}} \geq \gamma \cdot F_{LP}^*$, which concludes the proof. \blacksquare