

Toward Optimal Sniffer-Channel Assignment for Reliable Monitoring in Multi-Channel Wireless Networks

Dong-Hoon Shin^{*}, Saurabh Bagchi[†], and Chih-Chun Wang[†]

^{*}School of Electrical, Computer and Energy Engineering, Arizona State University, USA

[†]School of Electrical and Computer Engineering, Purdue University, USA

Email: donghoon.shin.2@asu.edu, {sbagchi, chihw}@purdue.edu

Abstract—This paper studies the optimal sniffer-channel assignment for reliable monitoring in multi-channel wireless networks. This problem concerns how to deploy certain sniffers in a network (and tune their channels) so that they can overhear and verify communication among the other nodes, referred to as normal nodes. Prior works have studied the optimal sniffer-channel assignment, but they assume perfect sniffers. However, in practice, sniffers may probabilistically make errors in monitoring, e.g., due to poor reception and compromise by an adversary. Hence, to maintain acceptable monitoring quality, a node needs to be overheard by multiple sniffers. We show that the optimal sniffer-channel assignment with sniffer redundancy differs fundamentally from the previous works due to the absence of a desirable property called submodularity. As a result, in our problem, the prior approximation algorithms no longer maintain their performance guarantees. We propose a variety of approximation algorithms based on two approaches—greedy strategy and relaxation-and-rounding approach. We present an empirical performance analysis of the proposed algorithms through simulations in practical networks. Our results suggest that our two algorithms show a performance trade-off between coverage and running time and are therefore suitable for different kinds of deployment.

I. INTRODUCTION

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

Over the past few years, it has been extensively studied to use multiple channels in wireless networks, especially in wireless mesh networks (WMNs) (e.g., [1]–[3]). 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 these networks, a challenging issue with passive monitoring is to capture as large an amount of traffic or as large a number of nodes as possible, ideally the

entire network, by judiciously assigning channels to sniffers’ radios. The model in all the works that are related to this paper is that the network is comprised of two kinds of nodes—sniffer nodes (“the monitors”) and normal nodes (“the monitored”).

In recent years, the issue of the optimal channel assignment of sniffers has received increasing attention from the literature. Our prior works [4], [5] have studied a problem of how to optimally place sniffers and assign their channels to monitor multi-channel WMNs. Chhetri *et al.* [6] have studied two models of sniffers that assume different capabilities of sniffers’ capturing traffic. The first, called the *user-centric* model, assumes that frame-level information can be captured so that activities of different users are distinguishable. The second, called the *sniffer-centric* model, assumes that only binary information is available regarding channel activities, i.e., whether some user is active in a specific channel near a sniffer. The works [4]–[6] assume that the prior knowledge of the topology and the channel usages of nodes is given to, or can be inferred by, sniffers. On the other hand, Arora *et al.* [7] 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. Also, some works [8], [9], including our work, have developed distributed solutions that can scale to large networks.

However, all of the aforementioned works assume that sniffers are perfect. This implies that once a node has at least one sniffer within its transmission range operating on the same channel, the node’s activity will be always monitored without any error. In practice, however, sniffers may intermittently, periodically, or permanently stop functioning or generate erroneous reports in monitoring. This may happen due to a variety of reasons including poor reception (due to packet collisions or poor channel conditions), compromise by an adversary, operational failure and sleep mode for energy saving. Such imperfect sniffers decrease the quality of monitoring, and eventually lead to a degradation of the network performance.

In this paper, we allow for imperfect sniffers that may probabilistically make errors in monitoring. In this scenario, we wish to still maintain the accuracy of the monitoring above a certain level. To this end, our approach is to provide sniffer redundancy to each node. That is, each node has to meet a coverage requirement, defined as the minimum number of sniffers required for reliably monitoring the node. In this

This material is based in part upon work supported by the National Science Foundation under Grant Number CNS-0831999. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The work by Dong-Hoon Shin reflected in this paper was done when he was a Ph.D. student at Purdue University.

approach, we are interested in the problem of how to assign a set of channels to sniffers' radios such that the coverage requirements of *all* nodes are satisfied. We refer to this problem as the *Full-Coverage Reliable Monitoring* (FCRM). However, we show that it is NP-hard to find any feasible solution to FCRM, with the complexity growing exponentially with the number of sniffers.

Many target applications of this problem have a considerable number of sniffers. Also, the problem has to be solved repeatedly whenever nodes move and their channel assignments change. Hence, it is a reasonable goal to find approximate solutions that will be agile and thus applicable to practical networks. We thus turn our attention to an optimization problem corresponding to FCRM, which is how to find a sniffer-channel assignment that maximizes the number (or the total weight) of nodes whose coverage requirement is met. We call this problem the *Maximum-Coverage Reliable Monitoring* (MCRM). It is also NP-hard, since one can find the answer to FCRM, an NP-hard problem, by solving its corresponding optimization problem, i.e., MCRM.

Our problem MCRM fundamentally differs from the previously studied problems in the literature [4]–[9] that assume perfect sniffers and thus do not need to consider sniffer redundancy. We show that a desirable property called *submodularity* no longer holds in MCRM. Submodularity is known as an important property in discrete optimization since it allows us to efficiently find provably (near-)optimal solutions, similar to convexity in continuous optimization. As a result, in MCRM-MC, the performance guarantees of the prior approximation algorithms do not apply.

In this paper, we propose a variety of approximation algorithms to solve MCRM based on two approaches—greedy strategy and relaxation-and-rounding approach. First, we develop two variants of a look-ahead greedy algorithm, which make a greedy decision looking a few steps ahead. Next, we develop two relaxations and two rounding algorithms, thus proposing four variants of a relaxation-and-rounding algorithm. We present an empirical performance analysis of the proposed algorithms through simulations in practical networks—random networks and scale-free networks—in terms of coverage and running time. Our results show that two algorithms based on the relaxation-and-rounding approach outperform the others; one achieves coverage comparable to the maximum coverage but with a relatively long running time, while the other attains coverage slightly lower than that of the former but at a lower running cost. Also, we empirically see that the structure of MCRM becomes more complicated as the coverage requirements of nodes increase by observing that the gap between the highest coverage achieved by our algorithms and the upper bound achieved by relaxation becomes larger.

II. PROBLEM FORMULATION

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 of nodes are chosen according to one of many

existing channel assignment algorithms in the literature (e.g., [1]–[3]). Each node n is given a coverage requirement r_n that is a positive integer and denotes the minimum number of sniffers required to reliably monitor node n . The value of the coverage requirements can be determined, e.g., as in [10], by using the failure model of sniffers (i.e., false positives and false negatives) and the desired accuracy of monitoring. We say that node n is *covered* if it is overheard by at least r_n sniffers operating on the same channel as the node. Also, each node n is given a non-negative weight w_n indicating the importance of monitoring the node n . These weights can be used to capture various application-specific objectives of monitoring. For example, one can assign a higher weight to a node transmitting a larger volume of data. Or, for security monitoring, one can assign the weights by taking into account the trustworthiness of nodes 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 needs to determine a wireless channel from C to tune its radio to. 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 overheard by sniffer s being tuned to channel c . We define a *sniffer-channel assignment* as a subset of \mathcal{K} that includes only one coverage-set for each sniffer. This constraint is due to the limitation on sniffers' radios that each sniffer has a single radio and can thus tune its radio to only one channel at a time.

A. Full-Coverage Reliable Monitoring

We first consider a decision problem to determine whether or not there exists a sniffer-channel assignment that achieves the full coverage, i.e., covers all nodes in N . We refer to this problem as the *Full-Coverage Reliable Monitoring* (FCRM). We denote the FCRM with k channels and a set $\vec{r} = (r_n : n \in N)$ of coverage requirements of nodes by FCRM(k, \vec{r}).

Theorem 1: FCRM(k, \vec{r}) is NP-hard, even for $k = 2$ and $\vec{r} = (2, 1, \dots, 1)$.

To prove the theorem, one can show a polynomial-time reduction from FCRM($2, \vec{1}$), which is an NP-hard problem [6, Theorem 1], to FCRM(k, \vec{r}) for any fixed k and \vec{r} . We omit the proof due to space limitation.

B. Maximum-Coverage Reliable Monitoring

Alternatively, we turn our attention to an optimization problem corresponding to FCRM, which is how to find a sniffer-channel assignment that maximizes the total weight of nodes whose coverage requirements are met. We refer to this problem as the *Maximum-Coverage Reliable Monitoring* (MCRM). We denote the MCRM with k channels and a set $\vec{r} = (r_n : n \in N)$ of coverage requirements of nodes by MCRM(k, \vec{r}). Also, we denote MCRM($k, \vec{1}$) with $k \geq 2$ by MCRM-SC (MCRM with Single Cover) and MCRM(k, \vec{r}) with $k \geq 2$ and $r_n \geq 2$ for some nodes $n \in N$ by MCRM-MC (MCRM with Multiple Cover).

The corollary below follows from Theorem 1, since one can find the answer to FCRM by solving MCRM and verifying whether the full coverage is achieved.

Corollary 1: MCRM(k, \vec{r}) is NP-hard, even for $k = 2$ and $\vec{r} = (2, 1, \dots, 1)$.

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

Corollary 2: For any $\epsilon > 0$, it is NP-hard to approximate MCRM(k, \vec{r}) within a factor of $\frac{7}{8} + \epsilon$ of the maximum coverage, even for $k = 2$ and $\vec{r} = (2, 1, \dots, 1)$.

Proof: Due to the proof of Theorem 1, it is easy to show that FCRM($2, \vec{1}$) is reduced to MCRM(k, \vec{r}). Also, it is NP-hard to approximate FCRM($2, \vec{1}$) within a factor of $\frac{7}{8} + \epsilon$ for any $\epsilon > 0$ [6, Corollary 2]. Thus, the corollary follows. ■

This implies that the best approximation ratio attainable for MCRM is at most $\frac{7}{8}$.

Non-submodularity of MCRM-MC. Submodularity is an important property in discrete optimization. It allows to efficiently find provably (near-)optimal solutions, similar to convexity in continuous optimization [11]. A real-valued function $f : 2^S \rightarrow \mathbb{R}$, defined on the subsets of a finite set S , is said *submodular* if and only if, for any $X \subseteq S - \{a\}$, the derived set function $\Delta f(a|X) \triangleq f(X \cup \{a\}) - f(X)$ is monotonically increasing, i.e., $\Delta f(a|X) \geq \Delta f(a|Y)$ for $X \subseteq Y$. Intuitively, submodularity is a *diminishing-return* property.

On the other hand, non-submodular functions are known to be difficult to deal with. In the literature of theoretical computer science, there are few results on the provable performance guarantees for non-submodular functions. Also, many greedy heuristics with good performance demonstrated in computational experiments cannot receive a theoretical analysis due to the difficulty of dealing with non-submodular functions [12]. We follow the same approach and give up the goal of trying to find approximation algorithms with provable performance guarantees. Instead, we focus on heuristic-based approximation algorithms that give good coverage empirically.

To investigate the submodularity on MCRM, we define $w : 2^{\mathcal{K}} \rightarrow \mathbb{R}$, defined on collections of coverage-sets in \mathcal{K} , to compute the total weight of the nodes covered by a sniffer-channel assignment. We first consider MCRM-SC. In MCRM-SC, a node is covered if it is overheard by at least one sniffer on the same channel as the node. Hence, the increment of the total weight by adding a coverage-set $K_{s,c}$ to a given sniffer-channel assignment \mathcal{A} , i.e., $\Delta w(K_{s,c}|\mathcal{A})$, is non-increasing as the given \mathcal{A} becomes a superset. Thus, w is submodular for MCRM-SC. Due to the submodularity of MCRM-SC, it is possible to approximate MCRM-SC within a factor of $1 - \frac{1}{e}$ (≈ 0.632) of the maximum coverage [9].

On the other hand, the submodularity no longer holds for

MCRM-MC.

Theorem 2: For MCRM-MC, w is not submodular.

Proof: We prove the theorem by a counter example. Assume that there exists a node $n \in N$ such that $r_n \geq 2$. We construct an instance of MCRM(k, \vec{r}) where $w_n = 1$ (i.e., the weight of node n is 1) and $K_{1,1} = \dots = K_{r_n,1} = \{n\}$ (i.e., sniffers $1, \dots, r_n$ can overhear only the node n by tuning their radios to channel 1). Consider two sniffer-channel assignments $\mathcal{A} = \emptyset$ and $\mathcal{A}' = \{K_{1,1}, \dots, K_{r_n-1,1}\}$. Then, it follows that $\Delta w(K_{r_n,1}|\mathcal{A}) = 0$ and $\Delta w(K_{r_n,1}|\mathcal{A}') = 1$. As a result, we have $\Delta w(K_{r_n,1}|\mathcal{A}) < \Delta w(K_{r_n,1}|\mathcal{A}')$ for $\mathcal{A} \subset \mathcal{A}'$. Thus, the theorem holds. ■

III. LOOK-AHEAD GREEDY ALGORITHMS

Our first approach to solve MCRM-MC is a greedy strategy. For MCRM-SC, an intuitive greedy algorithm has been presented in [4]. At each step, it picks the coverage-set that maximizes the coverage improvement, i.e., the total weight of uncovered nodes, among the coverage-sets of the sniffers whose channel assignment is not yet determined. This intuitive greedy algorithm can always achieve at least half of the maximum achievable coverage in MCRM-SC [4]. However, the performance guarantee of the greedy algorithm no longer holds for MCRM-MC due to the non-submodularity of MCRM-MC.

A. Naive Greedy Algorithms

To solve MCRM-MC, one might consider extending the greedy algorithm in [4]. We can think of two extensions of the greedy algorithm for MCRM-MC. At each step, among the coverage-sets of the sniffers whose channel assignment is not determined, the first extension picks the coverage-set that maximizes the coverage improvement at the step, while the second extension picks the coverage-set that maximizes the sum of the weights of the hitherto uncovered nodes, i.e., the nodes whose coverage requirements have not been met yet. Note that picking a hitherto uncovered node may still leave it uncovered. Note that, in MCRM-MC, these two greedy extensions yield a different solution, in general. To see this, observe that an uncovered node n in MCRM-MC can have a partial coverage of $1, \dots, r_n - 2$, or $r_n - 1$, other than zero coverage. Hence, when a coverage-set is picked at a step, only the nodes in the coverage-set that have the partial coverage of $r_n - 1$ will be covered.

However, these greedy extensions make poor choices due to their myopic nature, and this may lead to an inferior performance. To illustrate this, we consider the example shown in Table I. It is easy to see that the optimal sniffer-channel assignment is $\{K_{1,2}, K_{2,2}, K_{3,2}, K_{4,2}\}$, and thus the maximum achievable coverage is 5 nodes. On the other hand, the first greedy extension will pick a sniffer-channel assignment $\{K_{1,1}, K_{2,1}, K_{3,1}, K_{4,1}\}$ leading to a coverage of 2 nodes, provided that ties are broken by a coverage-set that achieves a higher total weight of uncovered nodes. Also, the second greedy extension will choose a sniffer-channel assignment $\{K_{1,1}, K_{2,2}, K_{3,2}, K_{4,1}\}$, thereby leading to zero coverage.

TABLE I
EXAMPLE TO ILLUSTRATE MYOPIC DECISIONS OF THE NAÏVE GREEDY
EXTENSIONS: $w_n = 1, r_n = 2$ FOR ALL n

Sniffers	Coverage-sets	
	Channel 1	Channel 2
s_1	$K_{1,1} = \{n_1, n_2, n_3, n_4\}$	$K_{1,2} = \{n_5, n_6, n_7\}$
s_2	$K_{2,1} = \{n_1\}$	$K_{2,2} = \{n_5, n_6, n_7\}$
s_3	$K_{3,1} = \{n_2\}$	$K_{3,2} = \{n_8, n_9, n_{10}\}$
s_4	$K_{4,1} = \{n_{11}, n_{12}, n_{13}\}$	$K_{4,2} = \{n_8, n_9\}$

B. Look-Ahead Greedy Algorithms

Inspired by the observation through the previous example, we design two look-ahead greedy algorithms to solve MCRM-MC, which are presented in Alg. 1 and Alg. 2.

Alg. 1 has a fixed number $|S|$ of steps. At each step, it looks $t' - 1$ steps ahead to find a coverage-set that is best for the current and the next $t' - 1$ steps. Here, t' is the minimum of the configuration parameter t and the number $|S'|$ of remaining steps of the algorithm. To find the best coverage-set, it first finds a collection \mathcal{C}^* of t' coverage-sets that maximize the coverage improvement for the current and the next $t' - 1$ steps (line 5). This takes $O(|S|^{t+1}|C|^{t+1})$ time, where t is a constant that we can choose. Then, among the coverage-sets in \mathcal{C}^* , it chooses a coverage-set K_{s^*,c^*} that maximizes the coverage improvement at the current step (lines 6 and 7).

Alg. 2 has a variable number of steps. At each step, it chooses a collection \mathcal{C}^* of at most t' coverage-sets that maximizes the *per-sniffer* coverage improvement, among all possible sniffer-channel assignments for any t' sniffers whose channel assignment is not yet determined (lines 4 and 5).

For both the algorithms, how far the algorithm can look ahead is determined by the parameter t . Since covering node n requires at least r_n sniffers, it is reasonable to set $t = \max_{n \in N} r_n - 1$ for the Look- t -Steps-Ahead Greedy Algorithm and $t = \max_{n \in N} r_n$ for the Look- t -Sniffers-at-One-Step Greedy Algorithm. If t is set to the largest value (i.e., $|S| - 1$ or $|S|$), the look-ahead greedy algorithms will solve MCRM-MC exactly. However, their computational complexity will grow exponentially with $|S|$, i.e., the number of sniffers.

IV. RELAXATION-AND-ROUNDING ALGORITHMS

We consider another approach, called relaxation and rounding (RaR), to solve MCRM-MC. RaR is known as a highly effective technique to solve NP-hard optimization problems. The typical steps involved in RaR algorithms are:

- Step 1: Formulate the given optimization problem into an integer program (IP)
- Step 2: Transform the IP into a relaxed program where the integer constraints are relaxed and that is solvable in polynomial time
- Step 3: Solve the relaxed program to obtain the optimal solution
- Step 4: Round the non-integer values from Step 3 in order to obtain an integer solution feasible for the original IP

At Step 2, an important issue is to find as strong a relaxed program as possible, while keeping the relaxed program solvable in polynomial time. We define a stronger relaxed

Algorithm 1 Look- t -Steps-Ahead Greedy Algorithm

- 1: $\mathcal{G} \leftarrow \emptyset, S' \leftarrow S$
- 2: **while** $|S'| \neq 0$ **do**
- 3: $t' \leftarrow \min\{t + 1, |S'|\}$
- 4: $\mathcal{P} \leftarrow \{\{K_{s_1, c_1}, \dots, K_{s_{t'}, c_{t'}}\} : s_i \in S', c_i \in C \forall i, \text{ and } s_i \neq s_j \text{ if } i \neq j\}$
// \mathcal{P} is the set of all possible sniffer-channel assignments of any t' sniffers in S'
- 5: Find $\mathcal{C}^* \in \mathcal{P}$ such that

$$\Delta w(\mathcal{C}^*|\mathcal{G}) = \max_{\mathcal{C} \in \mathcal{P}} \Delta w(\mathcal{C}|\mathcal{G})$$

// i.e., \mathcal{C}^* is a sniffer-channel assignment for any t' sniffers in S' that maximizes the coverage improvement, given \mathcal{G}

- 6: Find $K_{s^*, c^*} \in \mathcal{C}^*$ such that

$$\Delta w(\{K_{s^*, c^*}\}|\mathcal{G}) = \max_{K_{s, c} \in \mathcal{C}^*} \Delta w(\{K_{s, c}\}|\mathcal{G})$$

// i.e., K_{s^*, c^*} is a coverage-set in \mathcal{C}^* that maximizes coverage improvement, given \mathcal{G}

- 7: $\mathcal{G} \leftarrow \mathcal{G} \cup \{K_{s^*, c^*}\}$
- 8: $S' \leftarrow S' - \{s^*\}$
- 9: **end while**
- 10: **return** \mathcal{G}

Algorithm 2 t -Sniffers-at-One-Step Greedy Algorithm

- 1: $\mathcal{G} \leftarrow \emptyset$
- 2: **while** $|S'| \neq 0$ **do**
- 3: $\mathcal{Q} \leftarrow \{\{K_{s_1, c_1}, \dots, K_{s_{t'}, c_{t'}}\} : t' \leq \min\{t, |S'|\}, s_i \in S', c_i \in C \forall i, \text{ and } s_i \neq s_j \text{ if } i \neq j\}$
// i.e., \mathcal{Q} is the set of all possible sniffer-channel assignments for at most t' sniffers in S'
- 4: Find $\mathcal{C}^* \in \mathcal{Q}$ such that

$$\frac{\Delta w(\mathcal{C}^*|\mathcal{G})}{|\mathcal{C}^*|} = \max_{\mathcal{C} \in \mathcal{Q}} \frac{\Delta w(\mathcal{C}|\mathcal{G})}{|\mathcal{C}|}$$

// i.e., \mathcal{C}^* is a sniffer-channel assignment in \mathcal{Q} for at most t' sniffers that achieves the maximum *per-sniffer* coverage improvement

- 5: $\mathcal{G} \leftarrow \mathcal{G} \cup \mathcal{C}^*$
- 6: $S' \leftarrow S' - s(\mathcal{C}^*)$, where $s(\mathcal{C}^*)$ denotes the set of the sniffers chosen in \mathcal{C}^*
- 7: **end while**
- 8: **return** \mathcal{G}

program as follows. Let us say both programs P_1 and P_2 include the optimal IP solution, and program P_1 has a feasible solution set that is a subset of that of P_2 . Then P_1 will be called a stronger relaxed program than P_2 . The benefits of a stronger relaxed program are two folds. First, it often leads to a better approximate solution to the IP, since a stronger relaxed program will likely yield a non-integer solution closer to the optimal IP solution. Second, it will give a better estimate (i.e., upper bound) of the maximum coverage. At Step 4, a

challenging goal is to minimize the degradation of the quality of the resulting integer solution so as to obtain an integer solution that is as close to the optimal IP solution as possible.

A. LP and SDP Relaxations

We devise two relaxations; one is a linear program (LP) relaxation, and the other is a semidefinite program (SDP) relaxation. LP relaxation is a widely-used relaxation technique, but may not yield a tight upper bound for MCRM-MC. Hence, we also devise a SDP relaxation for MCRM-MC to obtain a stronger relaxation than the LP relaxation. To formulate the relaxations, we define a set of indicator variables. We assign an indicator variable $x_n \in \{0, 1\}$ to each node $n \in N$, and $x_n = 1$ 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 the radio of sniffer s is tuned to channel c .

LP relaxation. We formulate MCRM as the following integer linear program (ILP), denoted by ILP_{MCRM} :

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

$$\text{subject to} \quad \sum_{c \in C} y_{s,c} = 1 \quad \forall s \in S, \quad (2)$$

$$x_n \leq \frac{1}{r_n} \sum_{s,c: n \in K_{s,c}} y_{s,c} \quad \forall n \in N, \quad (3)$$

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

The constraint (2) is due to the fact that each sniffer has a single radio and the radio can be tuned to only one channel at a time. The constraints (3) together with the objective function (1) and the constraint (4) makes $x_n = 1$ if node n is covered by a solution, and otherwise makes $x_n = 0$.

We transform ILP_{MCRM} into the following LP relaxation, denoted by LP_{MCRM} :

$$\text{maximize} \quad \sum_{n \in N} w_n x_n \quad (5)$$

$$\text{subject to} \quad \sum_{c \in C} y_{s,c} = 1 \quad \forall s \in S, \quad (6)$$

$$x_n \leq \frac{1}{r_n} \sum_{s,c: n \in K_{s,c}} y_{s,c} \quad \forall n \in N, \quad (7)$$

$$0 \leq x_n, y_{s,c} \leq 1 \quad \forall n \in N, s \in S, c \in C, \quad (8)$$

$$x_n (|\{(s,c) : n \in K_{s,c}\}| - r_n) \geq 0 \quad \forall n \in N. \quad (9)$$

The integer constraint (4) in ILP_{MCRM} is relaxed to the fractional constraint (8).

One could have a naïve LP relaxation of Eqs. (5)–(8) by simply relaxing the integer constraint (4) in ILP_{MCRM} to the fractional constraint (8). But, to obtain a stronger LP relaxation, we additionally include the constraint (9) in LP_{MCRM} . Note that the additional constraint (9) along with the objective function (5) and the constraint (8) makes $x_n = 0$, when the number of sniffers within range of node n is smaller than

r_n . Hence, constraint (9) enforces the algorithm for solving LP_{MCRM} to give zero coverage to the nodes that are impossible to cover due to an insufficient number of sniffers that can overhear the nodes. We see empirically that this additional constraint improves the solution quality of the relaxed LP problem (by 9% of that of the naïve LP relaxation, on average).

SDP relaxation. We formulate MCRM into the quadratically constrained linear program, denoted by $\text{QCLP}_{\text{MCRM}}$:

$$\text{maximize} \quad \sum_{n \in N} w_n x_n \quad (10)$$

$$\text{subject to} \quad \sum_{c \in C} y_{s,c} = 1 \quad \forall s \in S, \quad (11)$$

$$x_n \left(\frac{1}{r_n} \sum_{s,c: n \in K_{s,c}} y_{s,c} - 1 \right) \geq 0 \quad \forall n \in N, \quad (12)$$

$$y_{s,c}(y_{s,c} - 1) = 0 \quad \forall s \in S, c \in C, \quad (13)$$

$$x_n(x_n - 1) = 0 \quad \forall n \in N. \quad (14)$$

The constraints (13) and (14) represent the integer constraints of x_n and $y_{s,c}$, respectively. The constraint (12) together with (10) and (14) makes $x_n = 1$ if node n is covered by a solution, and otherwise makes $x_n = 0$.

We now formulate a SDP relaxation from $\text{QCLP}_{\text{MCRM}}$. We define $\vec{z} = (x_1, \dots, x_{|N|}, y_{1,1}, \dots, y_{|S|,|C|})$ and denote the i -th entry of \vec{z} by z_i . We define a symmetric square matrix M of variables as

$$M = \begin{pmatrix} 1 & \vec{z} \\ \vec{z}^T & Z \end{pmatrix}.$$

Here, Z is a symmetric square matrix of new variables $Z_{i,j}$'s with order $|N| + |S| \cdot |C|$, and $Z_{i,j}$ denotes the entry in the i -th row and the j -th column of Z . To derive a SDP relaxation, we first add the constraints (7)–(9) in LP_{MCRM} to $\text{QCLP}_{\text{MCRM}}$ and rewrite the $\text{QCLP}_{\text{MCRM}}$ with the additional constraints (7)–(9) into the following matrix form:

$$\text{maximize} \quad W \bullet M \quad (15)$$

$$\text{subject to} \quad A_i \bullet M \leq b_i, \quad i \in I \quad (16)$$

$$Z = \vec{z}^T \vec{z}. \quad (17)$$

Here, W and A_i are symmetric square matrices of order $|N| + |S| \cdot |C|$, b_i is a real number, and I is an index set. The notation \bullet denotes the Frobenius inner product, i.e., $W \bullet M = \sum_{i,j} W_{i,j} M_{i,j}$, and $(\cdot)^T$ denotes the matrix transpose. The constraints (7)–(9) from LP_{MCRM} are added to obtain a stronger SDP relaxation. Note that, due to constraint (17), $Z_{i,j}$ is equal to the quadratic term $z_i z_j$. Following the standard procedure [13], we now transform Eqs. (15)–(17) into the following SDP relaxation, denoted by SDP_{MCRM} :

$$\text{maximize} \quad W \bullet M \quad (18)$$

$$\text{subject to} \quad A_i \bullet M \leq b_i, \quad i \in I \quad (19)$$

$$M \succeq 0 \quad (\Leftrightarrow Z - \vec{z}^T \vec{z} \succeq 0). \quad (20)$$

Here, $M \succeq 0$ means that the matrix M must be positive semidefinite, i.e., satisfy $\vec{v}M\vec{v}^T \geq 0$ for any real vector \vec{v} .

SDP_{MCRM} is a relaxed program of $\text{QCLP}_{\text{MCRM}}$ equivalent to ILP_{MCRM} , since a zero matrix is positive semidefinite and hence $Z - \vec{z}^T \vec{z} = 0$ implies $Z - \vec{z}^T \vec{z} \succeq 0$. In SDP_{MCRM} , $Z_{i,j}$ is no longer equal to $z_i z_j$ and now becomes an independent variable. Hence, SDP_{MCRM} is an LP defined over the variables in M , attached with the positive semidefinite constraint (20). Intuitively, we can interpret SDP_{MCRM} as a polynomial-time complexity emulation of $\text{QCLP}_{\text{MCRM}}$ by introducing the auxiliary variables $Z_{i,j}$'s and aiming $Z_{i,j} = z_i z_j$ with the constraints (12)–(14) and (20). Note that the objective function of SDP_{MCRM} is still defined over only the variables x_1, \dots, x_n , and also that the value that x_n can take is constrained by the constraints of LP_{MCRM} (i.e., Eqs. (6)–(9)). Hence, we have the following theorem.

Theorem 3: SDP_{MCRM} is a relaxation of ILP_{MCRM} that is at least as strong as LP_{MCRM} .

B. Randomized Rounding Algorithm and Greedy Rounding Algorithm

We develop two distinct rounding algorithms to round the non-integer values of $y_{s,c}$'s obtained from LP_{MCRM} or SDP_{MCRM} ; one is randomized, while the other is deterministic.

Randomized Rounding Algorithm (RRA). It probabilistically rounds the (fractional) LP_{MCRM} or SDP_{MCRM} optimal solution \vec{y}^* by treating each fractional value as the probability of rounding it to 1. That is, $P(y_{s,c}^\# = 1) = y_{s,c}^*$, where $y_{s,c}^\#$ denotes the resulting integer value of a fractional value $y_{s,c}^*$ after rounding by RRA.

Greedy Rounding Algorithm (GRA). We present GRA in Alg. 3. It rounds \vec{y}^* by iteratively choosing a sniffer-channel pair whose value will be rounded to 0. At each iteration (lines 4–16), for each sniffer-channel pair $p = (s, c)$ whose value is not yet rounded to an integer, GRA adjusts the values of $y_{s,1}^p, \dots, y_{s,|C|}^p$ according to Eq. (21). At the iteration, GRA finds the sniffer-channel pair $\tilde{p} = (\tilde{s}, \tilde{c})$ whose associated adjusted values $\vec{y}^{\tilde{p}}$ achieve the maximum coverage improvement (line 9). Here, the coverage improvement attained by $\vec{y}^{\tilde{p}}$, compared to $\vec{y}^\#$, is defined by $\Delta w(\vec{y}^{\tilde{p}}, \vec{y}^\#) = \sum_{n \in N(s)} (w_n(\vec{y}^{\tilde{p}}) - w_n(\vec{y}^\#))$, where $w_n(\vec{y}) = w_n \cdot \left\lfloor \min \left\{ 1, \frac{1}{r_n} \sum_{(s,c): n \in K_{s,c}} y_{s,c} \right\} \right\rfloor$, $N(s)$ is the set of the neighboring nodes of sniffer s , and $\lfloor x \rfloor$ denotes the largest integer that is not greater than x . Intuitively, the coverage improvement computes the coverage improvement attained by $\vec{y}^{\tilde{p}}$ as the total weight of the nodes that are newly covered by the adjusted fractional values $\vec{y}^{\tilde{p}}$. Then, at the end of the iteration, GRA updates the fractional values of sniffer \tilde{s} to the adjusted values of sniffer \tilde{s} in $\vec{y}^{\tilde{p}}$, thereby rounding at least one non-integer value to an integer (lines 10–15).

V. TIME COMPLEXITY ANALYSIS

A. Time Complexity of Look-Ahead Greedy Algorithms

The look-ahead greedy algorithms both have at most $|S|$ iterations of the while loop. At each iteration, Look- t -Steps-

Algorithm 3 Greedy Rounding Algorithm

```

1: Let  $\vec{y}^*$  be the optimal solution to  $\text{LP}_{\text{MCRM}}$  or  $\text{SDP}_{\text{MCRM}}$ 
2:  $\vec{y}^\# \leftarrow \vec{y}^*$ 
3:  $P \leftarrow \{p = (s, c) : 0 < y_{s,c}^\# < 1 \ \forall s \in S, c \in C\}$ 
4: while  $P \neq \emptyset$  do
5:   for each  $p = (s, c) \in P$  do
6:      $\vec{y}^p \leftarrow \vec{y}^\#$ 
7:     Adjust the values of the entries  $y_{s,1}^p, \dots, y_{s,|C|}^p$  of  $\vec{y}^p$ 
       according to:
           
$$y_{s,c}^p \leftarrow 0, \quad y_{s,c'}^p \leftarrow \frac{y_{s,c'}^p}{\sum_{c \in C} y_{s,c}^p} \ \forall c' \neq c \quad (21)$$

8:   end for
9:   Find  $\tilde{p} = (\tilde{s}, \tilde{c}) \in P$  that maximizes the coverage
       improvement gained by  $\vec{y}^{\tilde{p}}$  (i.e.,  $\Delta w(\vec{y}^{\tilde{p}}, \vec{y}^\#)$ )
10:   $\vec{y}^\# \leftarrow \vec{y}^{\tilde{p}}$ 
11:  if  $y_{\tilde{s},c}^{\tilde{p}} \in \{0, 1\}$  for all  $c \in C$  then
12:     $P \leftarrow P - \{(\tilde{s}, 1), \dots, (\tilde{s}, |C|)\}$ 
13:  else
14:     $P \leftarrow P - \{(\tilde{s}, \tilde{c})\}$ 
15:  end if
16: end while
17: return  $\vec{y}^\#$ 

```

Ahead Greedy Algorithm and t -Sniffers-at-One-Step Greedy Algorithm need to consider at most $O(|S|^{t+1}|C|^{t+1})$ possible sniffer-channel assignments in \mathcal{P} and at most $O(|S|^t|C|^t)$ possible sniffer-channel assignments in \mathcal{Q} , respectively. Here, t (i.e., the look-ahead capability) is assumed to be less than a half of $|S|$, which is true for almost all cases. Also, any sniffer-channel assignment has at most $O(|N|)$ nodes whose coverage needs to be verified to compute the coverage improvement. Thus, Look- t -Steps-Ahead Greedy Algorithm has time complexity of $O(|S|^{t+2}|C|^{t+1}|N|)$, and t -Sniffers-at-One-Step Greedy Algorithm has time complexity of $O(|S|^{t+1}|C|^t|N|)$.

B. Time Complexity of Relaxation-and-Rounding Algorithms

To compute the time complexity of the RaR algorithms, we first compute the time complexity of formulating and solving the LP_{MCRM} or the SDP_{MCRM} , and then compute the time complexity of GRA or RRA.

Formulating and solving LP_{MCRM} . To formulate LP_{MCRM} , we need to build an LP in the following matrix form: maximize $\vec{c} \cdot \vec{x}$ subject to $A\vec{x} = \vec{b}$ and $\vec{x} \geq 0$. In the formulation of LP_{MCRM} , building matrix A with the constraints (6)–(9) dominates the complexity, which will take $O((|N| + |S| \cdot |C|)^2)$ time since we have $|N| + |S| \cdot |C|$ variables and $O(|N| + |S| \cdot |C|)$ constraints in LP_{MCRM} . To solve LP_{MCRM} , one can employ one of many existing LP solvers, e.g., the one in [14] which will take $O((|N| + |S| \cdot |C|)^3 / \log(|N| + |S| \cdot |C|))$ time. Thus, in total, it takes $O((|N| + |S| \cdot |C|)^3 / \log(|N| + |S| \cdot |C|))$ time to formulate and solve LP_{MCRM} .

Formulating and solving SDP_{MCRM} . To formulate SDP_{MCRM} , constructing the matrices A_i 's in the constraint (19) dominates

TABLE II
TIME COMPLEXITY OF PROPOSED ALGORITHMS

Algorithm	Time complexity
Look- t -Steps-Ahead Greedy	$O(S ^{t+2} C ^{t+1} N)$
t -Sniffers-at-One-Step Greedy	$O(S ^{t+1} C ^t N)$
Solving LP _{MCRM}	$O\left(\frac{(N + S \cdot C)^3}{\log(N + S \cdot C)}\right)$
Solving SDP _{MCRM}	$O((N + S \cdot C)^3)$
RRA	$O(S \cdot C)$
GRA	$O(S ^2\cdot C ^2\cdot N)$

the complexity. This will take $O((|N| + |S| \cdot |C|)^3)$ time, since each A_i has $(|N| + |S| \cdot |C| + 1)^2$ entries and SDP_{MCRM} has $O(|N| + |S| \cdot |C|)$ constraints. To solve SDP_{MCRM}, one can use one of various SDP solvers available, which will take $O((|N| + |S| \cdot |C|)^3)$ time [15]. Thus, in total, it takes $O((|N| + |S| \cdot |C|)^3)$ time to formulate and solve SDP_{MCRM}.

Solving RRA and GRA. It is easy to compute that RRA and GRA have the time complexity of $O(|S| \cdot |C|)$ and $O(|S|^2 \cdot |C|^2 \cdot |N|)$, respectively.

Based on the results, we summarize the time complexity of the proposed algorithms in Table II.

VI. PRACTICAL IMPLEMENTATIONS

The two variants of the look-ahead greedy and the SDP-based RaR algorithms can be implemented by employing a centralized network entity, which first gathers from each sniffer the information of the channel usage of nodes, then runs the algorithm to determine the channel assignment of sniffers, and distributes the decision to every sniffer. This centralized setting would be suitable for networks whose configuration (i.e., nodes's channel-usage and network topology) changes slowly with time.

On the other hand, the LP-based RaR algorithms can be implemented in a distributed manner. One can develop a distributed algorithm to solve LP_{MCRM} by applying the methods in [9]. Also, it is easy to see that RRA and GRA can be implemented in a distributed fashion since they require only local information. This distributed implementation can quickly adapt to frequent network configuration changes due to nodes' mobility and the addition and replacement of sniffers.

VII. NUMERICAL EXPERIMENTS

We evaluate the performance of the proposed algorithms through MATLAB simulations in two kinds of networks: random networks and scale-free networks. In random networks, nodes and sniffers with receiving range r are randomly deployed in a 1×1 square area with a uniform distribution. In scale-free networks, nodes are deployed such that the probability $f(d)$ of a node having degree d follows a power law of the form of d^{-p} , i.e., the number of nodes with high degree decreases exponentially. We pick the nodes with highest degrees as sniffers so as to achieve higher coverage than picking them randomly. The rationale behind choosing these two kinds of networks is that the performance of the proposed algorithms will largely depend on the distribution of

node degree, and the two kinds of networks have a significant difference in the node-degree distribution.

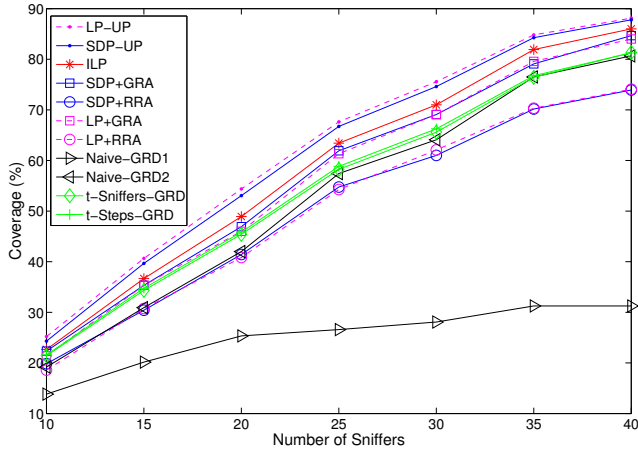
We evaluate the proposed algorithms in two metrics: coverage and running time. Here, coverage is defined as the total weight of nodes covered by a solution divided by the total weight of all nodes. We conduct two experiments in each network. In the first experiment, we see how the proposed algorithms perform as the number of sniffers increases. In the second experiment, we evaluate the proposed algorithms for different values of the coverage requirements of nodes. In the both experiments, each node's radio is tuned to a channel that is randomly selected from the available channels, and we set the parameters as follows: $r = 0.22$, $2 < p < 3$, $w_n = 1$ for all n , $|C| = 3$ (same as the number of non-overlapping wireless channels in IEEE 802.11), and $t = \max_v r_n - 1$ for Look- t -Steps-Ahead Greedy Algorithm and $t = \max_v r_n$ for t -Sniffers-at-One-Step Greedy Algorithm. All of the results are the average taken over at least 30 iterations.

A. Coverage

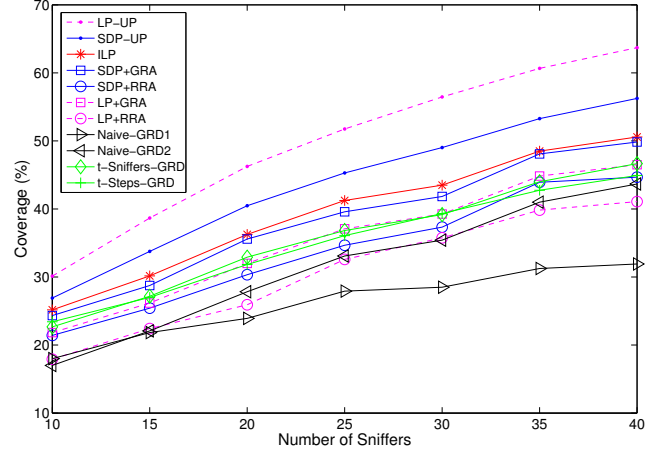
We compare the coverage of the proposed algorithms with the ILP optimum, i.e., the maximum achievable coverage, and the optimums of SDP_{MCRM} and LP_{MCRM}, denoted by SDP-UP and LP-UP respectively, which constitute upper bounds on the maximum coverage. In Fig. 1(a), we observe that the SDP-and-GRA and the LP-and-GRA show coverage comparable to the maximum achievable coverage, and they are followed by the look-ahead greedy algorithms with a small gap. We see that, after rounding, GRA maintains the quality of the optimal solution of SDP_{MCRM} or LP_{MCRM} closer to the maximum coverage, while RRA results in slight degradation of the solution quality. A surprising result is that the second naïve greedy algorithm shows reasonable coverage, while the first naïve greedy algorithm show poor coverage. Also, we observe that SDP_{MCRM} provides only a slightly tighter upper bound than LP_{MCRM}, and accordingly its corresponding RaR algorithms perform only slightly better than the LP-based algorithms.

In Fig. 1(b), we observe similar trends to those in random networks. But, a notable observation is that, in scale-free networks, SDP_{MCRM} provides a much tighter upper bound of the maximum achievable coverage. Accordingly, we can see that the SDP-based RaR algorithms show a noticeable coverage improvement over the LP-based RaR algorithms.

In Fig 2, we compare the coverage of the proposed algorithms only with upper bounds on the maximum coverage, i.e., SDP-UP and LP-UP, but not with the maximum coverage due to a large amount time to obtain the ILP optimum. We observe that, as r_n increases, the gap between the highest coverage achieved by SDP-and-GRA and the upper bound by SDP_{MCRM} becomes larger in both networks, and is larger in scale-free networks than in random networks. Also, we see that the first naïve greedy extension shows good coverage when $r_n = 1$, but the coverage dramatically decreases as r_n becomes 2. In both the figures, as r_n increases, the coverage decreases since there are a fixed number of sniffers and each node requires more number of sniffers to be covered.

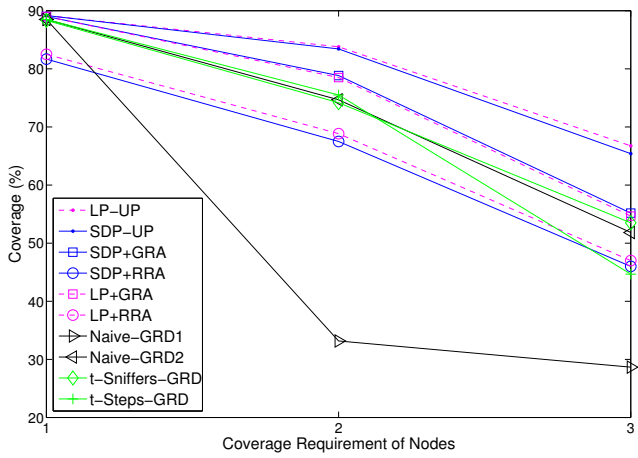


(a) Random network

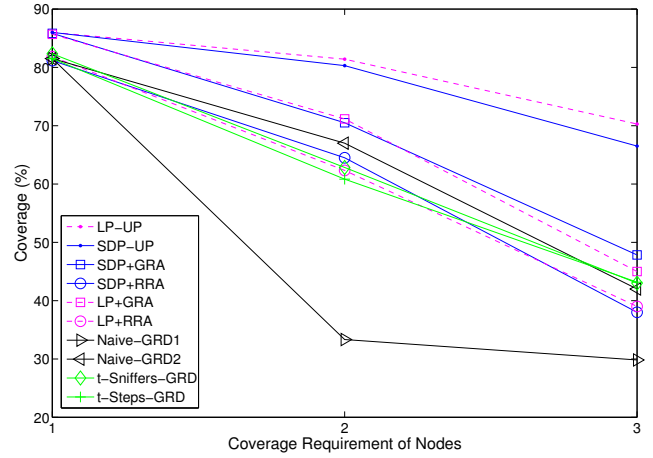


(b) Scale-free network

Fig. 1. Coverage for different values of $|S|$, where $|N| = 40$ and $r_n = 2$ for all n .



(a) Random network



(b) Scale-free network

Fig. 2. Coverage for $r_n = 1, 2$ and 3 for all n , where $|N| = 60$ and $|S| = 40$.

We would like to remark that, in practice, as the number of channels increases, sniffers need to be equipped with multiple radios to achieve a reasonably good coverage. We consider the extension to the multiple-radio case as our future work.

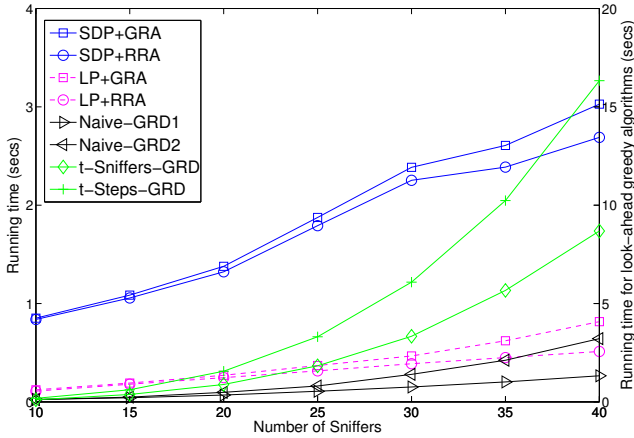
B. Running Time

Figure 3 shows the running time of the proposed algorithms in random networks. We present only the results for random networks since those for scale-free networks show similar results. In both the figures, there are two different y axes. The y axis on the right represents the running time of the look-ahead greedy algorithms, while the y axis on the left represents the running time of the other algorithms. The right y -axis shows a much larger time scale than the left y -axis, by 5 and 100 times for Fig. 3 (a) and (b), respectively.

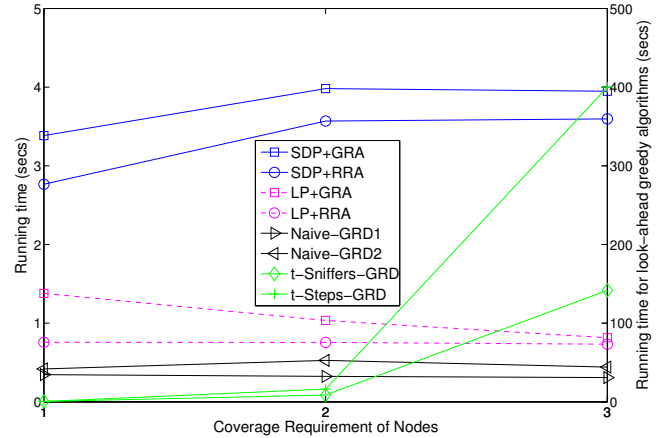
In Fig. 3 (a), we observe that the running times of the SDP-based RaR algorithms are substantially higher than those of

LP-based RaR algorithms, not expected from the asymptotic time complexity results. Also, we observe that the running times of the look-ahead greedy algorithms are much larger than those of the other algorithms, and they grow rapidly as the number of sniffers increases, as expected from its asymptotic time complexity of $|S|^3$. A notable observation is that the running time of t -Sniffers-at-One-Step Greedy Algorithm is almost half (or a third) of that of Look- t -Steps-Ahead Greedy Algorithm. This implies that t -Sniffers-at-One-Step Greedy Algorithm picks two (or three) coverage-sets at once for most of the iterations, while Look- t -Steps-Ahead Greedy Algorithm chooses only one coverage-set at every iteration.

Figure 3 (b) shows similar trends to those for the case of varying number of sniffers, in the comparison among the proposed algorithms. We observe that the naive greedy algorithms and the RaR algorithms show a relatively constant



(a) $|S|$ increases from 10 to 40, $|N| = 40$ and $r_n = 2$ for all n



(b) $r_n = 1, 2$ and 3 for all n , $|N| = 40$ and $|S| = 30$

Fig. 3. Running time for random networks

running time over different values of r_n , while the look-ahead greedy algorithms show a dramatically increasing running time as r_n increases, thus impractical for large values of r_n .

To summarize the simulation results, the SDP-and-GRA achieves the highest coverage close to the maximum coverage, but shows a (relatively) long running time. Hence, the SDP-and-GRA will be favored, especially, for monitoring applications where a higher coverage is more emphasized, such as security monitoring. On the other hand, the LP-and-GRA attains the coverage comparable to that of the SDP-and-GRA, but with a faster running time. Thus, LP-and-GRA can be considered as a good compromise between the coverage and the running-time, and will be favored for monitoring applications in dynamic network environments where the channel assignment of nodes changes rapidly.

VIII. CONCLUSION

In this paper, we studied the optimal sniffer-channel assignment problem for reliable monitoring in multi-channel wireless networks. In this, each node needs to be monitored by multiple sniffer nodes to maintain an acceptable monitoring quality. This problem fundamentally differs from the previously studied problems that assume perfect sniffers and thus do not need to consider sniffer redundancy. We proposed a variety of approximation algorithms based on two basic approaches—greedy and relaxation-and-rounding. We present a comparative analysis of the proposed algorithms through simulations. Our conclusion is that SDP-and-GRA achieves the highest coverage close to the maximum achievable coverage, but shows a (relatively) long running time. On the other hand, LP-and-GRA attains coverage comparable to that of the SDP-and-GRA at a lower running cost.

In future work, we will generalize the problem to the multiple-radio case where nodes and sniffers are equipped with multiple radios. Also, we will consider ways to come up with the degree of sniffer redundancy for various scenarios.

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*, August 2009.
- [4] D.-H. Shin and S. Bagchi, "Optimal Monitoring in Multi-Channel Multi-Radio Wireless Mesh Networks," in *Proc. of ACM MobiHoc*, May 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*, September 2010.
- [7] P. Arora, C. Szepesvari, and R. Zheng, "Sequential Learning for Optimal Monitoring of Multi-channel Wireless Networks," in *Proc. of IEEE INFOCOM*, April 2011.
- [8] 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.
- [9] D.-H. Shin, S. Bagchi, and C.-C. Wang, "Distributed Online Channel Assignment Toward Optimal Monitoring in Multi-Channel Wireless Networks," in *Proc. of IEEE INFOCOM, Mini-conference*, 2012.
- [10] D. Subhadrabandhu, S. Sarkar, and F. Anjum, "A Statistical Framework for Intrusion Detection in Ad Hoc Networks," in *Proc. of IEEE INFOCOM*, April 2006.
- [11] "Tutorials, references, activities and tools for submodular optimization:" [Online]. Available: <http://submodularity.org>
- [12] D.-Z. Du, R. L. Graham, P. M. Pardalos, P.-J. Wan, W. Wu, and W. Zhao, "Analysis of Greedy Approximations with Nonsubmodular Potential Functions," in *Proc. of ACM-SIAM SODA*, January 2008.
- [13] A. d'Aspersion and S. Boyd, "Relaxations and Randomized Methods for Nonconvex QCQPs," Stanford University, CA, EE392c Class Notes.
- [14] K. M. Anstreicher, "Linear Programming in $O(n^3/\ln n)$ Operations," *SIAM Journal on Optimization*, vol. 9, no. 4, pp. 803–812, 1999.
- [15] B. Borchers and J. G. Young, "Implementation of a primal-dual method for SDP on a shared memory parallel architecture," *Computational Optimization and Applications*, vol. 37, no. 3, pp. 355–369, July 2007.