Computer Communications xxx (2012) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Sniffer channel selection for monitoring wireless LANs

Xian Chen [a,*], Yoo-Ah Kim [b], Bing Wang [a], Yuan Song [a], Hieu Dinh [a], Guanling Chen [c]

[a] Computer Science & Engineering Department, University of Connecticut, United States
[b] National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bethesda, MD, United States
[c] Computer Science Department, University of Massachusetts, Lowell, United States

ARTICLE INFO

Article history:
Received 21 May 2011
Received in revised form 30 March 2012
Accepted 5 June 2012
Available online xxxx

Keywords:
WLAN
Sniffer channel selection
Monitor

ABSTRACT

Wireless sniffers are often used to monitor access points (APs) in wireless LANs (WLANs) for network management, fault detection, and traffic characterization. It is cost effective to deploy single-radio sniffers that can monitor multiple nearby APs. To achieve this, a sniffer needs to switch among multiple channels since these APs often operate on orthogonal channels. In this paper, we formulate and solve two optimization problems on sniffer channel selection. Both problems require that each AP be monitored by at least one sniffer. In addition, one optimization problem requires minimizing the maximum number of channels that a sniffer listens to, and the other requires minimizing the total number of channels that the sniffers listen to. We prove that both optimization problems are NP-hard. For each problem, we propose three algorithms to solve it, one based on integer programming (IP), one based on LP-relaxation, and the third based on a greedy heuristic. We evaluate the performance of the various algorithms using two real-world datasets. Our results show that, for each problem, all the three algorithms are effective in achieving their optimization goals, and overall, the LP-based algorithm outperforms the other two algorithms.

1. Introduction

Wireless LANs (WLANs) have been widely deployed in enterprise and campus networks. With this wide deployment, it becomes increasingly important to understand the behavior of WLANs, and automatically manage WLANs to ensure their normal operation and security. A widely-used and effective technique for understanding and monitoring WLANs is air sniffing, where a set of sniffers (also called air monitors, wireless monitors, or radio monitors) are placed inside a WLAN, each passively listening to the air waves in its vicinity, and collecting detailed MAC/PHY information (e.g., [31,19,12,20,14,13,23,21,10,9,25], more details in Section 2). Air sniffing has been shown to complement wire side monitoring that uses SNMP and base-station logs [31,19,14,15,10,9]. This is because the detailed MAC/PHY information (e.g., signal strength, spectrum density, collision, retransmissions, and backoff times) provides valuable insights into the behavior of wireless medium and protocols, which can help network administrators to optimize radio coverage and determine the root causes of network faults for effective trouble shooting. In addition, for mission critical WLANs with high security requirements, such as those deployed in banks or military bases [1], PHY/MAC visibility provided by the sniffers is critical, as wired solution can only detect upper layer threats.

While requiring additional infrastructure, the insights and benefits achieved by air sniffing cannot be obtained from traditional monitoring techniques (e.g., SNMP). On the other hand, large-scale WLAN monitoring through air sniffing faces several challenges. First, it requires a large number of sniffers, which can be costly to deploy and difficult to manage. This problem is compounded by the fact that access points (APs) in WLANs can operate on different channels (e.g., 802.11b/g supports 3 orthogonal channels, and 802.11a supports 13 orthogonal channels), while an air sniffer can only listen to a single channel at a given point of time (although a sniffer may use multiple radios to monitor multiple channels simultaneously, such type of sniffers are large and expensive to deploy [16]). Therefore, in the worst case, the required number of sniffers can be the same as the number of APs. Secondly, the sniffers generate a large amount of measurement data, which can be expensive to store, transfer and process. For instance, in [14], up to 80 Mbps of traffic is generated for monitoring an academic building, which needs to be transferred and processed at a central server.

The above challenges in large-scale air sniffing can be overcome by *channel sampling*, where each sniffer samples the network traffic by visiting multiple channels periodically [16]. Using channel sampling, a sniffer can monitor multiple nearby APs that operate on different channels, and hence less sniffers are needed. Furthermore, traffic sampling leads to less amount of measurement data.

* Corresponding author. Tel.: +1 860 486 1454; fax: +1 860 486 4817.
E-mail address: xian.chen@engr.uconn.edu (X. Chen).

0140-3664/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comcom.2012.06.005

As shown in [16], although not capturing all the traffic, channel sampling is useful for a number of applications, including security monitoring, anomaly detection, fault diagnosis, network characterization, and assistance to AP deployment.

In this paper, we address an important problem in channel sampling, namely, how to select channels for the sniffers to reduce the monitor cost. More specifically, we consider two problems. Both of them require that each AP be monitored by at least one sniffer, and in addition, one problem minimizes the maximum number of channels that a sniffer listens to, while the other minimizes the total number of channels that the sniffers listen to. Both optimization problems aim at reducing the number of channels that a sniffer needs to scan (in terms of worst case and average sense, respectively) since when a sniffer scans less channels, it can spend more time on each channel to improve sampling quality.[1] We prove that both optimization problems are NP-hard. For each problem, we propose three algorithms to solve it, one based on integer programming (IP), one based on LP-relaxation, and the third based on a greedy heuristic. We evaluate the performance of the various algorithms using two real-world datasets. Our results show that, for each problem, all the three algorithms are effective in achieving their optimization goals, and overall, LP-based algorithms outperform the other two algorithms.

The rest of the paper is organized as follows. Section 2 describes related work. Section 3 describes the problem setting. Sections 4 and 5 describe our sniffer channel selection algorithms for the two optimization problems, respectively. Section 6 presents performance evaluation. Finally, Section 7 concludes the paper and describes future work.

2. Related work

Several studies use air sniffing to understand and/or manage WLANs. Adya et al. [8] propose a client-based architecture that instruments wireless clients and (if possible) APs to monitor wireless medium and their nearby devices to detect and diagnose faults. Bahl et al. [11] propose using dense array of inexpensive radios (DAIR) through USB wireless adaptors that are attached to desktops to detect rogue wireless devices and Denial of Service attacks on WLANs. Later on, Chandra et. al. extend this DAIR architecture to incorporate location estimation and develop a location-based management system for WLANs [12]. Yan and Chen propose a model-based fault diagnosis approach that detects and localizes faults through self-monitoring at the APs [29]. Yeo et al. propose a framework that merges link-level measurement from multiple distributed air sniffers for WLAN management [31,32]. This framework is substantially extended in Jigsaw [14] and Wit [20], where the authors provide formal and systematic techniques to construct a global view of the network by merging and synchronizing traces from multiple locations. The global view has been used for understanding many aspects of WLANs, including congestion [19], link-layer losses and anomalies [20,31], co-channel interference [14], and sources of delays [13]. It has also been used to determine root-cause of physical-layer anomalies [23], and identify threats and attacks [21,22]. In addition to the above studies in academia, air sniffing has also been used in many commercial products (e.g., [2,4,5,3,7]). Our study uses air sniffers to monitor APs, and focuses on sniffer channel assignment, which has not been studied in these literatures.

As described earlier, air sniffing through dedicated sniffers can lead to high deployment cost and a large amount of monitoring traffic. The studies of [16,17] propose channel sampling to address the above two issues. In particular, the study of [16] proposes two sampling strategies, equal-time sampling where a sniffer spends equal amount of time scanning each channel, and proportional sampling where the amount of time that a sniffer spends on a channel is proportional to the amount of traffic on that channel. These two strategies are improved in [17] where the scanning of the sniffers are coordinated to increase the number of unique frames. Our study determines the set of channels that a sniffer scans during channel sampling. We require each sniffer to monitor a *subset* of selected channels, while [16,17] require each sniffer to monitor all available channels, regardless of whether the channels are being used or not by the nearby APs.[2] By eliminating the scanning over unused channels, our approach provides more effective traffic sampling. Several recent studies design centralized or distributed sniffer channel assignment algorithms [10,9,24,25]. These studies assume that the sniffers have multiple radios or assign channels to sniffers in a probabilistic manner, which differ from the context of channel sampling as in our study.

Table 1
Key notation.

Notation	Definition
V	Set of APs
C	Set of channels that the APs use
c_v	Channel that AP v uses, $v \in V,\ c_v \in C$
M	Set of sniffers
M_v	Set of sniffers that can hear the transmission from AP v
$\varphi(v)$	Assignment to AP v (the set of sniffers that monitor v)
$C_\varphi(m)$	Set of channels that sniffer m listens to based on assignments $\varphi(\cdot)$
M_φ	Set of sniffers that are used based on assignments $\varphi(\cdot)$

3. Problem setting

We now describe the problem setting. The key notation is summarized in Table 1 for easy reference. Consider a WLAN with a set of APs, V. Each AP uses a single radio, and hence a single channel, at any point of time (if an AP uses multiple channels simultaneously, we can regard it as multiple APs, each with a single channel). Let C denote the set of channels that the APs operate on. In particular, suppose AP v operates on channel c_v, $c_v \in C$. A set of sniffers (or monitors), M, is spread out in the WLAN to monitor the APs.[3] Let M_v denote the set of sniffers that are within the transmission range of v (i.e., M_v is the set of sniffers that can overhear the transmission of v when listening to channel c_v), $M_v \subseteq M$. We assume that $|M_v| \geqslant 1$, i.e., at least one sniffer can monitor v, $\forall v \in V$. Each sniffer has a single radio, and switches among multiple channels to monitor its nearby APs when these APs operate on different channels.

Motivated by what is adopted by commercial products (e.g., [5]), we assume that the WLAN uses a centralized management architecture, where a central controller manages the operation of the APs. The central controller knows the coordinates of the APs, and determines the channel for each AP. Furthermore, it knows the location of the sniffers, and determines the set of channels that each sniffer scans based on the locations of the APs and sniffers, and the channels of the APs. The PHY/MAC information collected by the sniffers is transmitted to the central controller for fault diagnosis and security analysis. This centralized architecture has many benefits: it reduces deployment and operating expenses, and

[1] We discuss tradeoffs of these two optimization problems in Section 3. In practice, a network administrator may choose to use one of these two optimization objectives based on the goals of the WLAN monitoring.

[2] One motivation of scanning all the channels in [16,17] is that it can capture rogue APs that operate on unused channels. Rogue APs, however, can be effectively detected using other approaches such as [28,30].

[3] The sniffers can be deployed as a separate infrastructure, or integrated on the APs themselves as in [29]. In this paper, for ease of exposition, we assume sniffers are deployed as a separate infrastructure.

Please cite this article in press as: X. Chen et al., Sniffer channel selection for monitoring wireless LANs, Comput. Commun. (2012), http://dx.doi.org/10.1016/j.comcom.2012.06.005

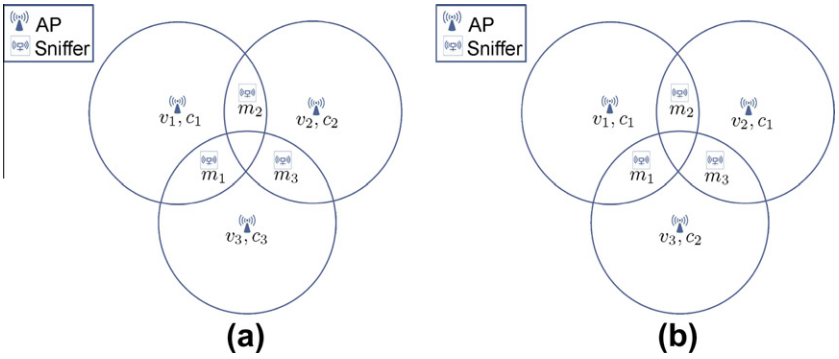

Fig. 1. Two examples to illustrate the problem setting. Both examples contain three APs, $v_1,\ v_2$ and v_3, and three sniffers, $m_1,\ m_2$, and m_3, where sniffer m_1 can monitor v_1 and v_3; sniffer m_2 can monitor v_1 and v_2; and sniffer m_3 can monitor v_2 and v_3. In (a) APs $v_1,\ v_2$, and v_3 use channels $c_1,\ c_2,\ c_3$, respectively; in (b) APs v_1 and v_2 use channel c_1, and AP v_3 uses channel c_2.

significantly simplifies daily operation and management of small to large-scale WLANs. Fig. 1 illustrates the problem setting using two examples. Both examples contain three APs, $v_1,\ v_2,\ v_3$, and three sniffers, $m_1,\ m_2,\ m_3$, that are controlled by the central controller. In addition, sniffer m_1 is in the transmission ranges of v_1 and v_3; sniffer m_2 is in the transmission ranges of v_1 and v_2; sniffer m_3 is in the transmission ranges of v_2 and v_3. They differ in the channels that the APs use: in Fig. 1(a), APs $v_1,\ v_2$, and v_3 use channels $c_1,\ c_2$, and c_3, respectively, while in Fig. 1(b), APs $v_1,\ v_2$ use channel c_1, and v_3 uses channel c_2.

Our goal is to determine the set of channels that each sniffer monitors. Let $\varphi(v)$ denote the set of sniffers that monitor AP v, referred to as *assignment* to v. Let $C_\varphi(m)$ denote the set of channels that sniffer m monitors based on the assignment $\varphi(\cdot)$. Then $C_\varphi(m) = \{c_v | m \in \varphi(v)\}$. Clearly, $C_\varphi(m) = \emptyset$, if $m \notin \varphi(v),\ \forall v \in V$. In this case, sniffer m is not used, and does not need to be deployed. We further define the *workload* of a sniffer as the number of channels that the sniffer scans. A sniffer is used if it monitors at least one channel, i.e., its workload is non-zero. Let $M_\varphi \subseteq M$ denote the set of sniffers that are being used. That is, $M_\varphi = \{m | C_\varphi(m) \neq \emptyset\}$.

We consider two sniffer channel selection problems. Both variants require that each AP be monitored by at least one sniffer, i.e., $\varphi(v) \neq \emptyset,\ \forall v \in V$. In addition, the first variant minimizes the maximum number of channels that a sniffer listens to (i.e., minimizes $\max_{m\in M}|C_\varphi(m)|$), while the second variant minimizes the sum of the channels that the sniffers listen to (i.e., minimizes $\sum_{m\in M}|C_\varphi(m)|$). We refer to these two variants as *min–max* and *min-sum sniffer channel selection problems*, respectively. In the min–max problem, the workloads of the sniffers are more balanced than those in the min-sum problem. On the other hand, the min-sum problem may need to use less sniffers and hence may have a lower deployment cost. The intuition comes from a special case: when there is a single channel, the min-sum problem minimizes the number of sniffers that need to be used. This is because, in this case, each sniffer needs to scan at most one channel, and hence minimizing the sum of the channels is equivalent to minimizing the number of sniffers that are used. Indeed, our extensive simulation demonstrates that the min-sum problem generally needs less sniffers than the min–max problem (see Section 6).

We further illustrate the difference of the min–max and min-sum problems using the two examples in Fig. 1. The optimal solutions of the two problems are the same for the example in Fig. 1(a), while differ for the example in Fig. 1(b). Specifically, for the example in Fig. 1(a), the optimal channel selection for both problems is making sniffers $m_1,\ m_2,\ m_3$ listen to channels $c_1,\ c_2,\ c_3$, respectively. For the example in Fig. 1(b), the optimal solution of the min–max problem is one, i.e., m_1 uses channel c_1 to monitor $v_1,\ m_2$ uses channel c_1 to monitor v_2, and m_3 uses channel c_2 to monitor v_3, which requires three sniffers, and the sniffer workloads are balanced. This channel selection leads to a solution of three for the min-sum problem, which is not optimal. The optimal solution of the min-sum problem is two, e.g., achieved as m_2 uses channel c_1 to monitor APs v_1 and v_2, and m_3 uses channel c_2 to monitor v_2, which only requires two sniffers, but the sniffer workloads are not balanced.

Last, neither optimization problem explicitly minimizes the number of sniffers that is being used (i.e., $|M_\varphi|$). Therefore, after solving the optimization problems, we post process the assignments to remove redundant sniffers to reduce the number of sniffers that are being used (see Sections 4 and 5).

4. Algorithms for min–max sniffer channel selection

We prove that the min–max sniffer channel selection problem is NP-hard by reducing 3-SAT (a known NP-complete problem) to it; the proof is found in Appendix A. In the following, we develop three algorithms to solve it. These three algorithms are based on integer programming (IP), linear programming (LP), and a greedy heuristic, referred to as *IP-min–max*, *LP-min–max*, and *Greedy-min–max*, respectively. We next describe the three algorithms in detail, and illustrate the results of solving the example in Fig. 1(a) (all three algorithms provide the same solution for the example in Fig. 1(b)). After obtaining a solution using one of the three algorithms, some redundant sniffers may be removed while still satisfying all the constraints. We therefore also propose a post-processing procedure that removes redundant sniffers and is applicable to all the three algorithms at the end of this section.

4.1. IP-min–max

Let $x_{m,c}$ be a 0–1 variable. In particular, $x_{m,c} = 1$ denotes that sniffer m monitors channel c, and $x_{m,c} = 0$ denotes otherwise. Then the min–max sniffer channel selection problem can be formulated as an IP problem:

$$\text{minimize}: \max_{m\in M} \sum_{c\in C} x_{m,c} \tag{1}$$

$$\text{subject to}: \sum_{m\in M_v} x_{m,c_v} \geqslant 1, \quad \forall v \in V \tag{2}$$

$$x_{m,c} \in \{0,1\} \tag{3}$$

In the objective function, $\sum_{c\in C} x_{m,c}$, is the total number of channels that sniffer m listens to, and Constraint (2) denotes that each AP is monitored by at least one sniffer.

For a small-scale problem, the above IP problem can be solved directly (e.g., using CPLEX [6]) to obtain an optimal solution for the

min–max problem. Afterwards, we determine the assignment for each AP, and the set of channels for each sniffer as follows. For AP v, let $\varphi(v) = \{m | m \in M_v, x_{m,c_v} = 1\}$. That is, we let all the sniffers that can overhear v and monitors c_v (i.e., the channel that v operates on) to monitor v. Correspondingly, we determine the set of channels that each sniffer monitors, i.e., $C_\varphi(m) = \{c | x_{m,c} = 1\}, \forall m$.

We now illustrate the results from IP-min–max using the example in Fig. 1(a). Solving the IP problem, we have $x_{m_i,c_i} = 1,\ i = 1,2,3$; others are 0, leading to an optimal solution of 1 for the min–max problem. For AP $v_i,\ i = 1,2,3$, we have $\varphi(v_i) = \{m_i\}$. And we have $C_\varphi(m_i) = c_i,\ i = 1,2,3$.

4.2. LP-min–max

For large problems, we may not be able to solve the IP problem in Section 4.1 directly. In our second algorithm, LP-min–max, we relax the integer constraint on $x_{m,c}$, and let $y_{m,c} \in [0,1]$ be the relaxed value of $x_{m,c}$. The original IP problem then becomes an LP problem, which can be solved in polynomial time. After solving the LP problem, we choose channels based on $y_{m,c}$ as described in Algorithm 1. In this algorithm, line 1 initializes $\varphi(v),\ C_\varphi(m)$ and M_φ to empty sets, $\forall v \in V,\ \forall m \in M$. Let S_v represent the set of APs that have already been considered. Line 2 initializes S_v to an empty set. The algorithm then considers all the APs. For an AP $v \in V$, if one monitor, m, has already been selected to be used and can hear v (i.e., $m \in M_\varphi \cap M_v$), and furthermore m has been assigned to monitor c_v (i.e., $c_v \in C_\varphi(m)$), then we simply assign m to monitor v. If there are multiple such monitors, all of them are recorded in $\varphi(v)$. If no such monitor exists, we pick a sniffer, m, that leads to the maximum y_{m,c_v} among all the sniffers that are in the transmission range of AP v (line 8). Once we add m to M_φ, the APs that have already been considered may also be monitored by m. Lines 12–16 find such APs, and assign m to monitor them as well. Line 18 updates S_v.

Algorithm 1. LP-min–max.

1: $\varphi(v) = \emptyset, \forall v \in V$, $C_\varphi(m) = \emptyset, \forall m \in M$, $M_\varphi = \emptyset$
2: $S_v = \emptyset$
3: **for all** $v \in V$ **do**
4: **for all** $m \in M_\varphi \cap M_v$ and $c_v \in C_\varphi(m)$ **do**
5: $\varphi(v) = \varphi(v) \cup \{m\}$
6: **end for**
7: **if** $\varphi(v) = \emptyset$ **then**
8: pick $m = \arg\max_{m \in M_v} y_{m,c_v}$
9: $C_\varphi(m) = C_\varphi(m) \cup \{c_v\}$
10: $\varphi(v) = \{m\}$
11: $M_\varphi = M_\varphi \cup \{m\}$
12: **for all** $v' \in S_v$
13: **if** $m \in M_{v'}$ and $c_{v'} = c_v$
14: $\varphi(v') = \varphi(v') \cup \{m\}$
15: **end if**
16: **end for**
17: **end if**
18: $S_v = S_v \cup \{v\}$
19: **end for**
20: Return $(\varphi, C_\varphi, M_\varphi)$

We next briefly describe the complexity of LP-min–max. The LP problem can be solved in polynomial time. In particular, when using interior point method, the running time is $O((|M||C|)^4)$, where $|M||C|$ is the number of variables. After solving the LP problem, the running time of Algorithm 1 to assign sniffer channels is $O(|V|)$. Therefore, the complexity of LP-min–max is $O((|M||C|)^4 + |V|) = O((|M||C|)^4)$.

We now illustrate LP-min–max using the example in Fig. 1(a). Solving the LP problem, we have one solution, $y_{m_1,c_1} = 0.5,\ y_{m_1,c_3} = 0.5,\ y_{m_2,c_1} = 0.5,\ y_{m_2,c_2} = 0.5,\ y_{m_3,c_2} = 0.5,\ y_{m_3,c_2} = 0.5$. Based on the values of $y_{m,c},\ \forall m \in M,\ c \in C$, we assign m_2 to monitor v_1 since both m_1 and m_2 can hear $v_1,\ y_{m_1,c_1} = y_{m_2,c_1}$.[4] Similarly, in the next iteration, we can assign m_2 to monitor v_2. Finally, we choose m_3 to monitor v_3. Therefore, we have a solution from LP-min–max as $C_\varphi(m_2) = \{c_1, c_2\}$ (m_2 uses channel c_1 and c_2), and $C_\varphi(m_3) = \{c_3\}$, leading to a suboptimal solution of 2 for the min–max problem. The assignment results are $\varphi(v_1) = \{m_2\}, \varphi(v_2) = \{m_2\}$, and $\varphi(v_3) = \{m_3\}$. Since $C_\varphi(m_1) = \emptyset,\ m_1$ is not used.

Last, the following theorem states an approximation-ratio result for LP-min–max; the proof is found in Appendix B.

Theorem 1. *LP-min–max is an $O(r)$-approximation algorithm for the min–max sniffer channel selection problem, where $r = \max_{v \in V} |M_v|$, i.e., r is the maximum number of sniffers that are in the transmission range of an AP.*

4.3. Greedy-min–max

Algorithm 2. Greedy-min–max.

1: $\varphi(v) = \{m | m \in M_v\},\ \forall v \in V$
2: $C_\varphi(m) = \emptyset$, $V_{m,c} = \emptyset, \forall m \in M,\ c \in C$
3: **for all** $v \in V$ **do**
4: **for all** $m \in M$ **do**
5: **if** $m \in M_v$ **then**
6: $C_\varphi(m) = C_\varphi(m) \cup \{c_v\}$
7: $V_{m,c_v} = V_{m,c_v} \cup \{v\}$
8: **end if**
9: **end for**
10: **end for**
11: $M_\varphi = M$
12: **repeat**
13: $M' = \emptyset$
14: **for all** $m \in M$
15: **if** $\exists c \in C_\varphi(m)$ s.t. $\forall v \in V_{m,c},\ |\varphi(v)| \geqslant 2$ **then**
16: $M' = M' \cup m$
17: **end if**
18: **end for**
19: **if** $M' \neq \emptyset$ **then**
20: Suppose $m \in M'$ monitors the largest number of channels
21: $C'_\varphi(m) = \{c | c \in C_\varphi(m), |\varphi(v)| \geqslant 2, \forall v \in V_{m,c}\}$
22: Pick $c \in C'_\varphi(m)$ that has the smallest $|V_{m,c}|$
23: $C_\varphi(m) = C_\varphi(m) \setminus \{c\}$
24: **if** $C_\varphi(m) = \emptyset$ **then**
25: $M_\varphi = M_\varphi \setminus \{m\}$
26: **end if**
27: $\varphi(v) = \varphi(v) \setminus \{m\},\ \forall v \in V_{m,c}$
28: $V_{m,c} = \emptyset$
29: **end if**
30: **until** M' is empty
31: Return$(\varphi, C_\varphi,\ M_\varphi)$

The main idea of Greedy-min–max is as follows. Initially, a sniffer, m, is assigned to monitor an AP, v, as long as m is in the transmission range of v. The algorithm then runs in iterations. In each iteration, it finds the sniffer with the maximum number of channels and removes one channel from this sniffer when feasible (i.e., while

[4] Here we break ties arbitrarily. Developing other approaches for breaking ties is left as future work.

Please cite this article in press as: X. Chen et al., Sniffer channel selection for monitoring wireless LANs, Comput. Commun. (2012), http://dx.doi.org/10.1016/j.comcom.2012.06.005

still satisfying the monitoring constraints) since our goal is to minimize the maximum number of channels that a sniffer uses. The iteration stops when none of the sniffers can remove any channel.

Algorithm 2 summarizes this algorithm. Line 1 initializes $\varphi(v)$ to be the set of sniffers that are in the transmission range of v, $\forall v \in V$. Let $V_{m,c}$ denote the set of APs that sniffer m monitors on channel c. Lines 2–10 initialize $C_\varphi(m)$ and $V_{m,c}$, $\forall m \in M$, $c \in C$. Line 11 initializes M_φ, the set of sniffers that are being used, to be the entire set of sniffers. In each iteration (lines 12–30), let M' record the set of sniffers that can remove at least one channel. It then picks a sniffer, m, that monitors the maximum number of channels from M'. Afterwards, it finds a channel, c, that can be removed and removes it from $C_\varphi(m)$. If multiple such channels exist, it chooses to remove the channel with the smallest number of APs (so that removing such a channel may affect the least number of APs). If after removing the channel, $C_\varphi(m)$ becomes an empty set, then we remove the sniffer m from M_φ (lines 24 to 26). Last, line 27 removes m from the assignment of all the APs in $V_{m,c}$ (since m does not monitor channel c any more), and line 28 sets $V_{m,c}$ to an empty set.

We now briefly describe the complexity of Greedy-min–max. The running time of lines 3–10 is $O(|V||M|)$. In each iteration of loop between lines 12–30, the algorithm picks one channel. So, in the worst case, the running time of this loop is $O(|M||C|)$. The sub-loop between line 14–18 can be finished within $O(|M||V||C|)$. We can neglect the running time of line 19–29. In summary, the running time of Greedy-min–max is $O(|M|^2|C|^2|V|)$.

We now illustrate Greedy-min–max using the example in Fig. 1(a). Initially, we assign each sniffer to monitor two channels at the beginning. In the iteration, we first set $M' = \{m_1, m_2, m_3\}$, and choose m_1, and then remove channel 1 from $C_\varphi(m_1)$. Set $\varphi(v_1) = \{m_2\}, \varphi(v_2) = \{m_2, m_3\}$, and $\varphi(v_3) = \{m_1, m_3\}$. In the second iteration, we choose m_2, and remove channel 2 from $C_\varphi(m_2)$. Then set $\varphi(v_2) = \{m_3\}, \varphi(v_3) = \{m_1, m_3\}$. In the last iteration, we choose m_3, and remove channel 3 from $C_\varphi(m_3)$. Therefore, $\varphi(v_3) = \{m_1\}$. In summary, the Greedy-min–max solution is $C_\varphi(m_1) = \{c_3\}$, $C_\varphi(m_2) = \{c_1\}$, and $C_\varphi(m_3) = \{c_2\}$. The assignment results are $\varphi(v_1) = \{m_2\}$, $\varphi(v_2) = \{m_3\}$, and $\varphi(v_3) = \{m_1\}$. This is also an optimal assignment, while it differs from the optimal solution from IP-min–max.

4.4. Remove redundant Sniffers

None of the above three algorithms explicitly minimizes the number of sniffers that are being used. As a result, the solutions may contain a large number of redundant sniffers. We next propose two algorithms to remove redundant sniffers. One is an optimal algorithm, based on IP, and the other is a heuristic algorithm that has much shorter running time.

The main idea of the IP-based algorithm is as follows. Given a sniffer channel selection solution $\varphi(\cdot)$, it exhaustively searches among the sniffers that are being used, and finds the maximum number of sniffers that can be removed while still maintaining that all the APs are being monitored by at least one sniffer. Specifically, the IP formulation is

$$\text{minimize} : z \tag{4}$$

$$\text{subject to} : \sum_{m \in \varphi(v)} z_{m,c_v} \geqslant 1, \quad \forall v \in V \tag{5}$$

$$\sum_{m \in M_\varphi} z_{m,c_m} \leqslant z, \quad \forall c_m \in C_{\varphi(m)}, \quad m \in M_\varphi \tag{6}$$

$$z_{m,c} \in \{0,1\}, \quad \forall m \in M, \quad \forall c \in C \tag{7}$$

$$z_{m,c} = 0, \quad \forall m \notin M_\varphi, \quad \text{or} \quad c \notin C_\varphi(m) \tag{8}$$

In this formulation, we define an integer variable z that represents the number of sniffers that are being used, and a set of binary variables $z_{m,c}$, where $z_{m,c} = 1$ indicates that sniffer m listens channel c, and $z_{m,c} = 0$ indicates otherwise, $m \in M$, $c \in C$. The objective function is minimizing z. The value of $z_{m,c}$ depends on the channel assignment solution $\varphi(\cdot)$: it is zero if sniffer m is not used or channel c is not in $C_\varphi(m)$; otherwise, $z_{m,c}$ can be either 0 or 1 (see constrains (7) and (8)). Constraint (5) requires the solution provided by $z_{m,c}$ covers all of the APs. Since a sniffer is used if it monitors at least one channel, the sum of $z_{m,c}$ for $m \in M_\varphi$, $c \in C_{\varphi(m)}$ is no less than the number of sniffers that is used. The set of constraints in (6) lists all possible such summations (there are $\Pi_{m \in M_\varphi} |C_\varphi(m)|$ such summations).

We now apply the above IP-based algorithm to remove redundant sniffers in the solutions provided by IP-min–max, LP-min–max and Greedy-min–max for the example in Fig. 1(a). For the solution from IP-min–max, constraint (6) is $z_{m_1,c_1} + z_{m_2,c_2} + z_{m_3,c_3} \leqslant z$; for the solution from LP-min–max, constraint (6) is $z_{m_2,c_1} + z_{m_3,c_3} \leqslant z$ and $z_{m_2,c_2} + z_{m_3,c_3} \leqslant z$; and for the solution from Greedy-min–max, constraint (6) is $z_{m_1,c_3} + z_{m_2,c_1} + z_{m_3,c_2} \leqslant z$. For all three solutions, solving the IP formulation (4)–(8), we find no sniffer can be removed.

The IP Algorithm needs exponential running time, and hence is not applicable to large-scale problems. We next propose a greedy algorithm, Algorithm 3, that has a polynomial running time to remove redundant sniffers. In Algorithm 3, $V_{m,c}$ denotes the set of APs that are monitored by sniffer m on channel c. Lines 1–6 initialize $V_{m,c}$ based on the sniffer channel assignment. Lines 7–14 consider all the sniffers that have been used, and for each sniffer, it checks all the channels that have been selected for the sniffer, and removes unnecessary channels (a channel is not necessary if all the APs are still monitored by at least one sniffer after removing this channel). At the end, lines 15–19 remove all the unnecessary sniffers (i.e., sniffers that do not monitor any channel) from M_φ.

Algorithm 3. Remove redundant sniffers (using a greedy heuristic).

```
1: V_{m,c} = ∅, ∀m ∈ M_φ, ∀c ∈ C
2: for all v ∈ V do
3:   for all m ∈ φ(v) do
4:     V_{m,c_v} = V_{m,c_v} ∪ {v}
5:   end for
6: end for
7: for all m ∈ M_φ do
8:   for all c ∈ C_φ(m) do
9:     if ∀v ∈ V_{m,c}, ∃m' ≠ m s.t. v ∈ V_{m',c}
10:      C_φ(m) = C_φ(m) \ {c}
11:      φ(v) = φ(v) \ {m}, ∀v ∈ V_{m,c}
12:    end if
13:  end for
14: end for
15: for all m ∈ M
16:   if C_φ(m) = ∅ then
17:     M_φ = M_φ \ {m}
18:   end if
19: end for
```

The running time of Algorithm 3 is $\sum_{v \in V} |\varphi(v)| + \sum_{m \in M_\varphi} |C_\varphi(m)|$. It may not provide optimal solutions. We compare the performance of the two algorithms that remove redundant sniffers in Section 6.

5. Algorithms for min-sum sniffer channel selection

It is easy to see that the min-sum sniffer channel selection problem is NP-hard. This is because when there is a single channel, it is

Please cite this article in press as: X. Chen et al., Sniffer channel selection for monitoring wireless LANs, Comput. Commun. (2012), http://dx.doi.org/10.1016/j.comcom.2012.06.005

equivalent to the minimum set cover problem, which is NP-hard. We also develop three algorithms to solve it, based on IP, LP-relaxation, and a greedy heuristic, referred to as *IP-min-sum*, *LP-min-sum*, and *Greedy-min-sum*, respectively. After running each algorithm, we can again use the algorithms in Section 4.4 to remove redundant sniffers. We next describe the three algorithms in detail.

IP-min-sum differs from IP-min–max in that it first solves an IP problem with the objective function

$$\text{minimize}: \sum_{m\in M}\sum_{c\in C} x_{m,c} \tag{9}$$

instead of (1) as in IP-min–max. LP-min-sum differs from LP-min–max in that it first solves an LP-relaxation problem for the min-sum problem instead of the min–max problem. LP-min-sum has the same complexity as LP-min–max. We have a similar approximate ratio result for LP-min-sum, as stated in the following theorem. The proof is found in Appendix B.

Theorem 2. *LP-min-sum is an $O(r)$-approximation algorithm for the min-sum sniffer channel selection problem, where $r = \max_{v\in V}|M_v|$, i.e., r is the maximum number of sniffers that are in the transmission range of an AP.*

Algorithm 4. Greedy-min-sum.

1: $\varphi(v) = \emptyset,\ \forall v \in V$, $C_\varphi(m) = \emptyset,\ \forall m \in M$, $M_\varphi = \emptyset$
2: $V_{m,c} = \emptyset, \forall m \in M,\ c \in C$
3: **for all** $v \in V$ **do**
4: **for all** $m \in M$ **do**
5: **if** $m \in M_v$ **then**
6: $V_{m,c_v} = V_{m,c_v} \cup \{v\}$
7: **end if**
8: **end for**
9: **end for**
10 $V' = V$
11: **repeat**
12: pick $m,\ c$ such that $|V_{m,c}| = \max_{m'\in M, c'\in C}|V_{m',c'}|$
13: $\varphi(v) = \varphi(v) \cup \{m\},\ \forall v \in V_{m,c}$
14: $C_\varphi(m) = C_\varphi(m) \cup \{c\}$
15: $M_\varphi = M_\varphi \cup \{m\}$
16: $V_{m',c} = V_{m',c} \setminus V_{m,c},\ \forall m' \in M$
17: $V' = V' \setminus V_{m,c}$
18: **until** V' is empty
19: Return $(\varphi, C_\varphi, M_\varphi)$

Greedy-min-sum models the sniffer channel selection problem as a minimum set covering problem: we map each sniffer to $|C|$ *virtual sniffers*, each monitoring one channel in C, then the min-sum problem is equivalent to finding the minimum number of virtual sniffers so that all APs are monitored and the number of virtual sniffers (and hence the sum of the channels used by all the sniffers) is minimized. Many algorithms have been proposed for the minimum set covering problem. Greedy-min-sum follows a greedy algorithm for minimum set covering problem [18]. It runs in iterations. In each iteration, it picks a sniffer and channel pair that can monitor the maximum number of APs. The iteration continues until all the APs are monitored.

Algorithm 4 summarizes this algorithm (we used a similar algorithm for scheduling sniffers to detect rogue APs in [30]). Let $V_{m,c}$ denote the set of APs that sniffer m could monitor if it listened to channel c. Line 1 initializes $C_\varphi(m)$, M_φ and $\varphi(v)$ to be empty sets, $\forall m \in M,\ v \in V$. Lines 2–9 initialize $V_{m,c},\ \forall m \in M,\ c \in C$. Line 10 initializes, V', the set of APs that have not been monitored, to V. The algorithm then runs in iterations until V' is empty. Using a greedy strategy, line 12 chooses a monitor and channel pair, m and c, that covers the maximum number of APs, i.e., $|V_{m,c}| = \max_{m'\in M, c'\in C}|V_{m',c'}|$ (if multiple such sniffers exist, we choose the one that monitors the minimum number of channels, i.e., with the minimum $|C_\varphi(m)|$). After choosing the monitor and channel pair, m and c, line 13 assigns m to all the APs in $V_{m,c}$; line 14 adds channel, c, into $C_\varphi(m)$; and line 15 adds monitor m to M_φ. Afterwards, since the APs in $V_{m,c}$ have already been monitored, line 16 removes $V_{m,c}$ from $V_{m',c}, \forall m' \in M$, and line 17 removes $V_{m,c}$ from V'.

Following the results in [18], the approximation ratio of Greedy-min-sum is H_d for the min-sum problem, where $H_d = \sum_{i=1}^{d} 1/i$ is the d-th harmonic number, and d is the maximum number of APs that a sniffer can monitor in its neighborhood. The complexity of Greedy-min-sum is $O(|M|^2|C|^2)$: the dominant complexity is in the loop between line 11–18; inside the loop, each iteration chooses one monitor m and one channel c, leading to $O(|M||C|)$ iterations, and inside an iteration, line 12 has running time of $O(|M||C|)$, and hence the total running time is $O(|M|^2|C|^2)$.

Last, we describe the assignment results when using the three algorithms to solve the example in Fig. 1(b) (all three algorithms obtain the same solution for the example in Fig. 1(a), details omitted in the interest of space). When using IP-min-sum, we have $x_{m_1,c_1} = 0,\ x_{m_1,c_2} = 1,\ x_{m_2,c_1} = 1,\ x_{m_2,c_2} = 0,\ x_{m_3,c_1} = 0$, and $x_{m_3,c_2} = 0$, leading to an optimal solution of two. Specifically, the assignment is $\varphi(v_1) = \varphi(v_2) = \{m_2\},\ \varphi(v_3) = \{m_1\}$, and hence $C_\varphi(m_1) = \{c_2\},\ C_\varphi(m_2) = \{c_1\}$, and $C_\varphi(m_3) = \emptyset$. When using LP-min-sum, we have $y_{m_1,c_1} = 0.5,\ y_{m_2,c_1} = 0.5,\ y_{m_3,c_1} = 0.5,\ y_{m_3,c_2} = 0.5,\ y_{m_1,c_2} = 0.5,\ y_{m_2,c_2} = 0$. Therefore, when selecting sniffer to monitor channel 1, y_{m_1,c_1}, y_{m_2,c_1}, and y_{m_3,c_1} are all 0.5.[5] Suppose we choose m_1, the solution is $\varphi(v_1) = \{m_1\},\ \varphi(v_2) = \{m_3\},\ \varphi(v_3) = \{m_1\}$, and hence $C_\varphi(m_1) = \{c_1, c_2\},\ C_\varphi(m_3) = \{c_1\}$, and $C_\varphi(m_2) = \emptyset$, leading to a suboptimal solution of 3. The solution obtained by Greedy-min-sum is the same as that by IP-min-sum. We again apply the algorithms in Section 4.4 to remove redundant sniffers, and find no sniffer can be removed for the solutions provided by the three algorithms.

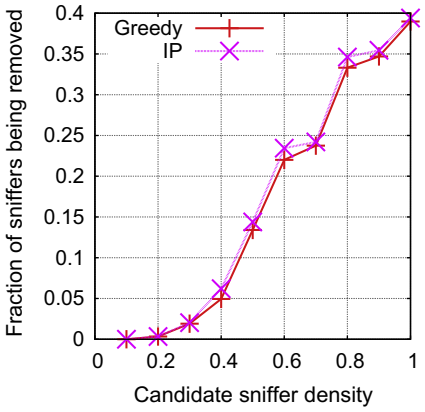

Fig. 2. Performance of the two algorithms that remove redundant sniffers. The results are for an area with 25 APs in the Dartmouth dataset. The sniffer channel selection solution is obtained using IP-min–max.

6. Performance evaluation

Our performance evaluation uses two empirical datasets. One corresponds to the campus WLAN network in Dartmouth College. The other is obtained using Placelab[6] from Seattle downtown area.

[5] We again break ties arbitrarily, as in the LP-min–max algorithm.
[6] http://www.placelab.org/database/.

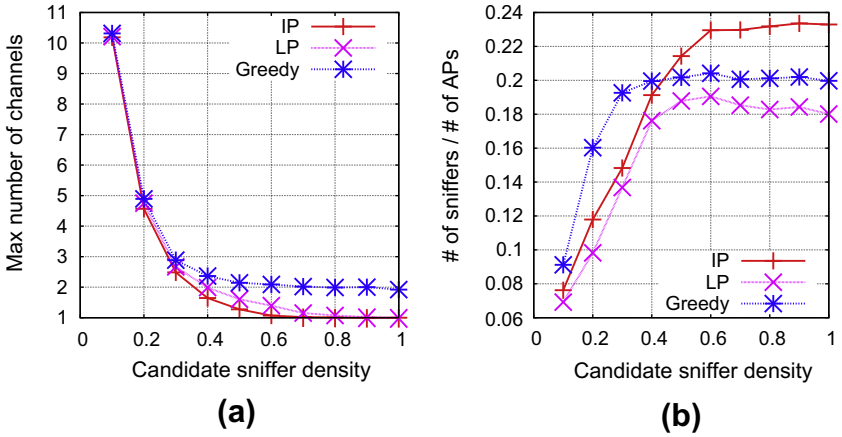

Fig. 3. The min–max sniffer channel selection problem: (a) maximum number of channels that a sniffer monitors, and (b) ratio of the number of sniffers that are being used over the number of APs. These results are for the 400-AP area in the Dartmouth dataset; IP, LP and Greedy are abbreviations for IP-min–max, LP-min–max, and Greedy-min–max, respectively.

Both datasets are obtained by wardriving. The APs are deployed in buildings, and can be densely deployed at certain locations. The Dartmouth dataset represents a well-managed wireless network, while the Seattle dataset is a unmanaged network. The Dartmouth dataset contains both AP location (2D coordinates) and channel information. More specifically, the APs in the dataset use both 802.11b/g and 802.11a, and operate on 12 orthogonal 2.4 GHz/5 GHz channels. In the following, we treat each AP as two duplicate APs, each working on one channel. The Seattle dataset only contains AP location information. We randomly assign each AP one channel from the 24 available channels for 802.11b/g and 802.11a. The transmission range of an AP in both datasets is set to 100 m.

We first evaluate the two algorithms that remove redundant sniffers (see Section 4.4). Since the IP-based algorithm cannot solve large-scale problems in a reasonable amount of time, we use a small network to compare these two algorithms. Specifically, we choose an area that contains 25 APs in the Dartmouth dataset. For systematic evaluation, we generate 1000 topologies by virtually placing candidate sniffers uniformly randomly into the area. The number of candidate sniffers is randomly chosen from 1 to the number of APs. For each topology, we obtain a pair (n_a, n_s), where n_a is the number of APs that can be monitored by at least one sniffer, and n_s is the number of sniffers that can monitor at least one AP (i.e., sniffers that are within the transmission range of at least one AP). Therefore n_a and n_s can be smaller than the number of APs and sniffers in the area, respectively. We refer to the ratio, n_s/n_a, as *candidate sniffer density*. For each topology, we run the IP-min–max algorithm to select channels for the sniffers, and then apply the two algorithms to remove redundant sniffers. Fig. 2 plots the fraction of sniffers that are removed versus candidate sniffer density, n_s/n_a. The results are aggregated over a bin size of 0.1, i.e., the result under $n_s/n_a = x$ is the average of all the topologies with $n_s/n_a \in (x-0.1, x]$ (the confidence intervals are tight and hence omitted). The results from both the IP-based optimal algorithm and the greedy heuristic are plotted in the figure. We observe that the performance of the greedy heuristic is close to that of the IP-based optimal algorithm. Considering the running time, the rest of the results in this section uses the greedy heuristic to remove redundant sniffers.

We now present the results when solving the min–max and min-sum problems in large-scale networks. For both the Dartmouth and Seattle datasets, we consider two 500 m × 500 m areas: one has approximately 400 APs, representing an area with densely deployed APs; the other has a much lower AP density (approximately 200 APs). To systematically evaluate the performance of our algorithms, for each area we consider, we again generate

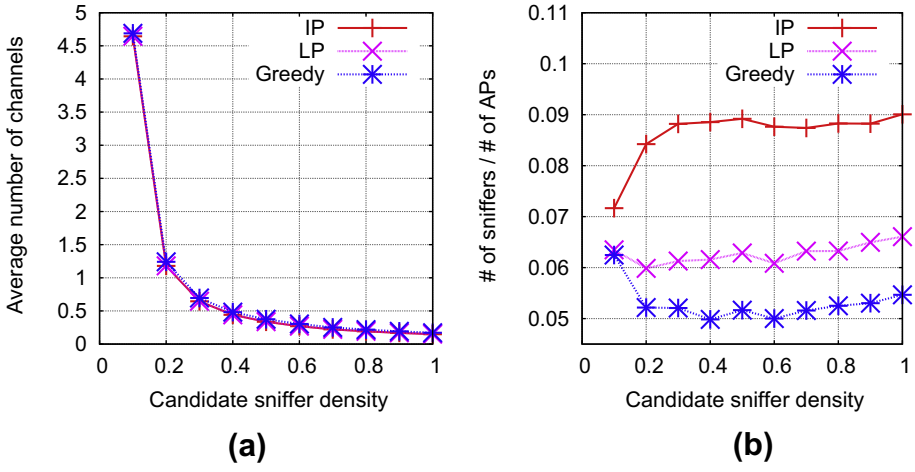

Fig. 4. The min-sum sniffer channel selection problem: (a) average number of channels that a sniffer monitors, and (b) ratio of the number of sniffers that are being used over the number of APs. These results are for the 400-AP area in the Dartmouth dataset; IP, LP and Greedy are abbreviations for IP-min-sum, LP-min-sum, and Greedy-min-sum, respectively.

Please cite this article in press as: X. Chen et al., Sniffer channel selection for monitoring wireless LANs, Comput. Commun. (2012), http://dx.doi.org/10.1016/j.comcom.2012.06.005

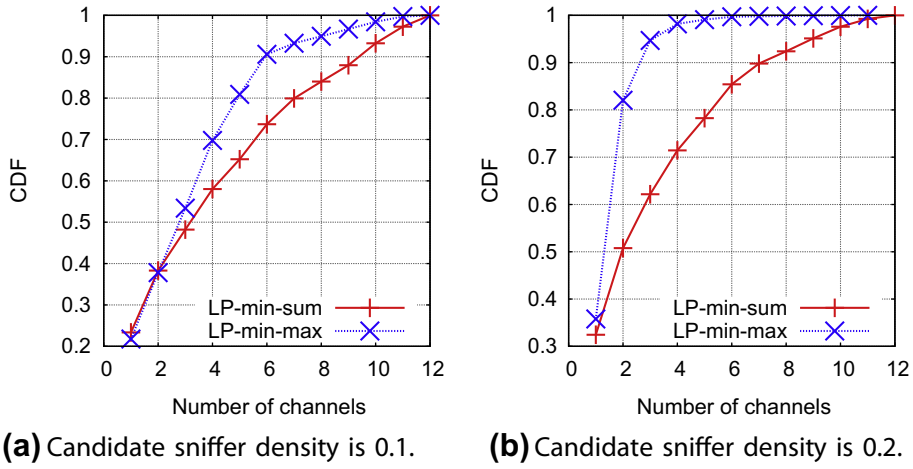

Fig. 5. Sniffer workload distribution when solving the min–max and min-sum sniffer channel selection problems for the 400-AP area in the Dartmouth dataset.

1000 topologies by virtually placing candidate sniffers uniformly randomly into the area.

Our simulation runs on a Intel Xeon PC with four 3.0 GHz processors. For each algorithm, the running time for the Dartmouth dataset is shorter than that for the Seattle dataset. This might be because the former uses 12 channels while the latter uses 24 channels. For both the min–max and min-sum problems, the LP-based algorithms are the fastest: it only takes a few minutes to finish solving all the 1000 topologies. The IP-based algorithms are the slowest: it can take up to 7 h to solve the 1000 topologies. The running time of the greedy heuristics is in between (it takes tens of minutes to finish the 1000 topologies). In the following, we mainly present the results for the 400-AP area in the Dartmouth dataset; results under other settings (the other area in the Dartmouth dataset and the two areas in the Seattle dataset) are similar.

Fig. 3 plots the results when solving the min–max sniffer channel selection problem for the 400-AP area in the Dartmouth dataset. The results under all the three algorithms, IP-min–max, LP-min–max, and Greedy-min–max, are plotted in the figure. Fig. 3(a) plots the maximum number of channels that a sniffer monitors versus candidate sniffer density, n_s/n_a. The results are again aggregated over a bin size of 0.1 (the confidence intervals are tight and hence omitted). We observe that for all three algorithms, as expected, the maximum number of channels used by the sniffers reduces as the candidate sniffer density increases. IP-min–max provides the optimal solution (in terms of the objective function). LP-min–max performs slightly worse than IP-min–max: the performances of these two algorithms are similar under both low and high sniffer densities; the difference is most noticeable for medium range of sniffer density (between 0.4 and 0.6). Both IP-min–max and LP-min–max outperform Greedy-min–max, particularly for large values of sniffer density. We also observe a diminishing gain from increasing the density of sniffers: the maximum number of channels decreases dramatically first and then less dramatically afterwards. Fig. 3(b) plots the ratio of the number of sniffers that are being used over the number of APs. We observe that LP-min–max outperforms the other two algorithms: the fraction of the sniffers that are being used under LP-min–max is 10% to 20% lower than that under IP-min–max, and is 5% to 25% lower than that under Greedy-min–max. Taking account of both the objective function and the number of sniffers needed, LP-min–max is a preferred algorithm than the other two: the maximum number of channels under LP-min–max is only slightly larger under the optimal solution, while the number of sniffers needed by LP-min–max is significantly lower than the other two algorithms. Last, from Fig. 3(a) and (b), we observe that for the min–max sniffer channel selection problem, a preferred candidate sniffer density is between 0.2 to 0.3, which leads to significant reduction in the maximum number of channels used by the sniffers compared to lower densities, while leads to moderate cost in deploying the air sniffing infrastructure: the number of sniffers that are being used is below 14% of the number of APs under LP-min–max, the preferred algorithm.

Fig. 4 plots the results for the min-sum sniffer channel selection problem when using the three algorithms, IP-min-sum, LP-min-sum, and Greedy-min-sum. Fig. 4(a) plots the average number of channels that a sniffer monitors versus candidate sniffer density, n_s/n_a. Each data point is the average calculated over all sniffers, excluding those that are not being used. We observe that all three algorithms lead to similar performance, both LP-min-sum and Greedy-min-sum provide solutions close to the optimal solution from IP-min-sum. Again, we observe a diminishing gain from increasing the density of sniffers. Fig. 4(b) plots the ratio of the number of sniffers that are being used over the number of APs. We observe Greedy-min-sum slightly outperforms LP-min-sum, and significantly outperforms IP-min-sum (it requires 13% to 40% less sniffers than IP-min-sum). For all the setttings, the number of sniffers that are being used is much smaller than the number of APs (the former is 5% to 9% of the latter), indicating a moderate cost of deploying the air sniffing infrastructure.

Summarizing the above observations, we conclude that, considering running time, the objective function, and the fraction of sniffers that are used, the LP-based algorithm outperforms the IP-based and the greedy-heuristic based algorithms for both the min–max and min-sum problems, considering both the running time and performance, LP-based algorithms and hence is a preferable choice for large network in practice.

Comparing Fig. 3(b) and Fig. 4(b), we see that, for the same candidate sniffer density, the min-sum problem requires much less sniffers than the min–max problem. For instance, when the sniffer density is 0.2, the ratio of the number of sniffers over the number of APs for the min–max problem is between 0.1 to 0.16, (see Fig. 3(b)), while the ratio for the min-sum problem is between 0.05 to 0.085 (see Fig. 4(b)). This is not very surprising: the min–max problem needs a larger number of sniffers since it requires the workloads of the sniffers to be balanced (to achieve the min–max goal). We next further compare the sniffer workload distribution for the min–max and min-sum problems. Fig. 5 plots the CDF (cumulative distribution function) of the sniffer workloads (i.e., workload of a sniffer is the number of channels that is used by

Please cite this article in press as: X. Chen et al., Sniffer channel selection for monitoring wireless LANs, Comput. Commun. (2012), http://dx.doi.org/10.1016/j.comcom.2012.06.005

the sniffer), excluding those sniffers that are not used. Again, it is for the 400-AP area in the Dartmouth dataset. We only plot the results under the LP-based algorithms (i.e., LP-min–max and LP-min-sum) when the sniffer density is 0.1 and 0.2 (shown in Fig. 5(a) and (b), respectively). From Fig. 5(a), we observe respectively 90% and 74% of the sniffers scan at most 6 channels in the min–max and min-sum problems. The difference is more dramatic in Fig. 5(b) where nearly 100% of the sniffers scan at most 4 channels in the min–max problem while the corresponding value is only around 70% in the min-sum problem.

7. Conclusions and future work

In this paper, we studied sniffer channel selection for monitoring WLANs. In particular, we formulated min–max and min-sum sniffer channel selection problems, and proposed three algorithms, one based on IP, one based on LP-relaxation, and the third based on a greedy heuristic, to solve each problem. Through simulation, we demonstrated that for each problem, all the algorithms are effective in achieving their optimization goals, and overall, the LP-based algorithm outperform the other two algorithms.

As future work, we plan to investigate dynamic sniffer channel assignment [26] that adjusts the channel assignment in the face of faults or attacks. Furthermore, our min–max and min-sum problems consider the number of channels that a sniffer monitors as the workload of the sniffer. Another direction of future work is using the amount of traffic that a sniffer monitors as the workload. Last, our current study assumes that the sniffer locations are known beforehand. An interesting direction is how to place the sniffers optimally given the network topology.

Acknowledgments

A preliminary version of this work appeared in [27]. This work was partially supported by NSF CAREER award 0746841, QSI Inc., UConn Faculty Large Grant, and the Science and Technology Directorate of the U.S. Department of Homeland Security under Award NBCH2050002.

Appendix A. Proof that min–max sniffer channel assignment is NP-hard

We prove that the min–max sniffer channel assignment problem is NP-hard by reducing 3-SAT to it.

Proof. Let ϕ be an instance of 3-SAT problem. Suppose ϕ contains n variables, $x_1,\ldots,x_n$, $2n$ literals, $x_1,\bar{x_1},\ldots,x_n,\bar{x_n}$, and m clauses, $C_1,\ldots,C_m$. The corresponding sniffer channel assignment problem contains $2n$ sniffers, $x_1,\bar{x_1},\ldots,x_n,\bar{x_n}$, and $(2m+2n)$ APs, $C_1,\bar{C_1},\ldots,C_m,\bar{C_m}$, and $D_1,\bar{D_1},\ldots,D_n,\bar{D_n}$. Each sniffer can operate on two channels, 1 and 2. Each AP operates on one channel. In particular, AP C_i operates on channel 1, and AP $\bar{C_i}$ operates on channel 2, $i=1,\ldots,m$; AP D_i operates on channel 1, and AP $\bar{D_i}$ operates on channel 2, $i=1,\ldots,n$. If x_i is used in clause C_j in ϕ, then in the corresponding sniffer channel assignment problem, sniffer x_i can monitor AP C_j using channel 1, and sniffer $\bar{x_i}$ can monitor AP $\bar{C_j}$ using channel 2. Similarly, if $\bar{x_i}$ is used in clause C_j in ϕ, then in the corresponding sniffer channel assignment problem, sniffer $\bar{x_i}$ can monitor AP C_j using channel 1, and sniffer x_i can monitor AP $\bar{C_j}$ using channel 2. Last, we allow and only allow sniffers x_i and $\bar{x_i}$ to monitor APs D_i and $\bar{D_i}$.

Fig. A.1 shows an example illustrating the relationship between ϕ and the corresponding sniffer channel assignment problem. In the example, $\phi=(x_1\vee\bar{x_2}\vee x_3)\wedge(\bar{x_1}\vee\bar{x_2}\vee\bar{x_3})$. We represent the relationship between the literals and the clauses in Fig. A.1(a),

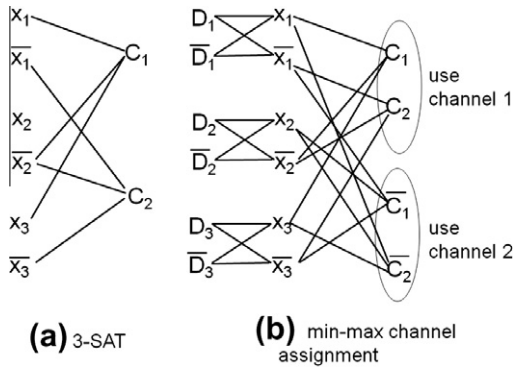

Fig. A.1. Illustration of the reduction from 3-SAT to the min–max sniffer channel assignment problem.

where a clause is connected to a literal if it contains that literal. Fig. A.1(b) shows the corresponding sniffer channel assignment problem. For instance, Fig. A.1(a) shows that x_1 is used in clause C_1 in ϕ. Correspondingly, Fig. A.1(b) shows that sniffer x_1 uses channel 1 to monitor AP C_1, and sniffer $\bar{x_1}$ uses channel 2 to monitor AP $\bar{C_1}$. For ease of illustration, in Fig. A.1(b), the upper circle contains $C_1,\ldots,C_m$ that use channel 1; and the lower circle contains $\bar{C_1},\ldots,\bar{C_m}$ that use channel 2. We further have x_i connects to D_i and $\bar{D_i}$, and $\bar{x_i}$ connects to D_i and $\bar{D_i}$.

We next show that there is a satisfying assignment to ϕ iff the solution to the min–max sniffer channel assignment problem is 1, i.e., each sniffer needs to scan at most one channel. We first prove that when ϕ is satisfiable, then each sniffer needs to scan at most one channel. Suppose that there exists an assignment to x_i, $i=1,\ldots,n$, so that all the clauses in ϕ are true, and hence ϕ is true. Consider an arbitrary clause C_j. Since C_j is true, at least one literal used in C_j must be true. We consider the following two cases:

- Case 1. Suppose a literal in C_j, x_i, is true. Then in the sniffer channel assignment problem, we let sniffer x_i monitor channel 1 (and hence it can monitor AP C_j), and sniffer $\bar{x_i}$ monitor channel 2 (and hence it can monitor AP $\bar{C_j}$). Under this channel assignment, both APs C_j and $\bar{C_j}$ are monitored.
- Case 2. Suppose a literal $\bar{x_k}$ in C_j is true (i.e., x_k is false). Then in the sniffer channel assignment problem, we let sniffer x_k monitor channel 2, and sniffer $\bar{x_k}$ monitor channel 1. Then again both APs C_j and $\bar{C_j}$ are monitored.

Summarizing the above two cases, we can find sniffer channel assignment for $x_i, i=1,\ldots,n$, so that all C_j's and $\bar{C_j}$'s are monitored, and each sniffer needs to monitor at most one channel. For APs D_i and $\bar{D_i}$, since only x_i and $\bar{x_i}$ are allowed to monitor them, and x_i and $\bar{x_i}$ monitor two different channels, it is easy to see that both of them can be monitored based on the current sniffer channel assignments. More specifically, if x_i monitors channel 1 (i.e., $\bar{x_i}$ monitors channel 2), we let x_i monitor D_i and $\bar{x_i}$ monitor $\bar{D_i}$ (since D_i and $\bar{D_i}$ operate on channels 1 and 2, respectively); if x_i monitors channel 2, we let x_i monitor $\bar{D_i}$ and $\bar{x_i}$ monitor $D_i, i=1,\ldots,n$. In summary, when ϕ is satisfiable, we can find channel assignments to all the sniffers so that each sniffer needs to monitor at most one channel and all the APs are monitored. Therefore, the solution to the min–max problem is 1.

We now prove that if each sniffer needs to scan at most one channel in the min–max sniffer channel assignment problem, then ϕ is satisfiable. Suppose there exists a channel assignment to all the sniffers, so that all the APs are monitored and each sniffer monitors

at most one channel. Consider all the sniffer pairs $(x_k, \bar{x}_k)$, $k = 1, \ldots, n$. Let $S_{i,j}$ denote the set of sniffer pairs in which sniffer x_k scans channel i and $\bar{x}_k$ scans channel $j, i, j = 1, 2$. We then have $S_{1,1} = S_{2,2} = \emptyset$. This is because, since D_i and $\bar{D}_i$ operate on two different channels and both of them are monitored, and in addition only x_i and $\bar{x}_i$ are allowed to monitor them, we must have x_i and $\bar{x}_i$ monitor two different channels, $i = 1, \ldots, n$. Hence we have $S_{1,1} = S_{2,2} = \emptyset$. In this case, for the APs to be monitored, we need $S_{1,2} \neq \emptyset$ and/or $S_{2,1} \neq \emptyset$. Consider an arbitrary sniffer pair $(x_i, \bar{x}_i)$. Then we have the following two scenarios:

- Sniffers x_i and $\bar{x}_i$ monitor channels 1 and 2, respectively. Then in ϕ, we set x_i to be true and set $\bar{x}_i$ to be false. Since x_i monitors channel 1, it can only monitor APs that use channel 1 (i.e., APs in the upper circle in Fig. A.1(b)). Suppose x_i monitors AP C_j. Then by the mapping between the 3-SAT and the min–max problem, x_i is used in clause C_j in ϕ, and therefore C_j is satisfied.
- Sniffers x_i and $\bar{x}_i$ monitor channels 2 and 1, respectively. Then in ϕ, we set the x_i to be false and set $\bar{x}_i$ to be true. Following a similar argument as before, suppose $\bar{x}_i$ monitors AP C_j, then by the mapping between the 3-SAT and the min–max problem, $\bar{x}_i$ is used in clause C_j in ϕ, then C_j is satisfied.

Summarizing the above two scenarios, since the sniffers in $S_{1,2} \cup S_{2,1}$ monitor all the APs, all the clauses in ϕ are satisfied. Hence we conclude that ϕ is satisfiable. Therefore, we have proved that if each sniffer needs to monitor at most one channel in the sniffer channel assignment problem, then there exists a satisfying assignment for ϕ, and hence complete the proof. □

Appendix B. Proof of approximation ratio for LP-based algorithms

We now prove Theorems 1 and 2 that state the approximate ratio of LP-min–max and LP-min-sum, respectively.

Proof. Consider an arbitrary AP, v, and a sniffer $m \in M_v$. Our LP rounding guarantees that $x_{m,c_v} \leqslant r y_{m,c_v}$, where $r = \max_{v \in V} |M_v|$. This can be shown by considering the following two cases. When $y_{m,c_v} = \max_{m \in M_v} y_{m,c_v}$, by our LP rounding, $x_{m,c_v} = 1$, and we have $x_{m,c_v} \leqslant r y_{m,c_v}$ (since $y_{m,c_v} \geqslant 1/r$). When $y_{m,c_v} \neq \max_{m \in M_v} y_{m,c_v}$, by our LP rounding, $x_{m,c_v} = 0 \leqslant r y_{m,c_v}$. Since the above AP, v, is chosen arbitrarily, we have

$$\sum_{c \in C} x_{m,c} \leqslant r \sum_{c \in C} y_{m,c}, \quad \forall m \in M.$$

Let n_m^* represent the optimal solution to the min–max sniffer channel selection problem. We have

$$\max_{m \in M} \sum_{c \in C} x_{m,c} \leqslant r \left(\max_{m \in M} \sum_{c \in C} y_{m,c} \right) \leqslant r n_m^*. \tag{B.1}$$

The second inequality above is because the LP provides a lower bound to the original problem. From (B.1), LP-min–max is an $O(r)$-approximation algorithm for the min–max sniffer channel selection problem. Similarly, let n_s^* represent the optimal solution to the min-sum problem. We have

$$\sum_{m \in M} \sum_{c \in C} x_{m,c} \leqslant r \left(\sum_{m \in M} \sum_{c \in C} y_{m,c} \right) \leqslant r n_s^*. \tag{B.2}$$

Hence LP-min-sum is an $O(r)$-approximation algorithm for the min-sum problem. □

References

[1] <http://csrc.nist.gov/publications/nistpubs/800-153/sp800-153.pdf>.
[2] AirDefense, Wireless LAN Security. <http://airdefense.net>
[3] AirMagnet. <http://www.airmagnet.com>.
[4] AirWave, AirWave Management Platform. <http://airwave.com>.
[5] Cisco Wireless LAN Solution Engine (WLSE). <http://www.cisco.com/en/US/products/sw/cscowork/ps3915/>.
[6] Cplex. <http://www.ilog.com/products/cplex/>.
[7] NetStumbler. <http://www.netstumbler.com>
[8] A. Adya, V. Bahl, R. Chandra, L. Qiu, Architecture and techniques for diagnosing faults, in: IEEE 802.11 infrastructure networks, in: Proc. of ACM MobiCom, September 2004.
[9] P. Arora, C. Szepesvari, R. Zheng, Sequential learning for optimal monitoring of multi-channel wireless networks, in: Proc. of IEEE INFOCOM, 2011.
[10] C. Arun, N. Huy, S. Gabriel, Z. Rong, On quality of monitoring for multi-channel wireless infrastructure networks, in: Proc. of ACM MobiHoc, 2010.
[11] P. Bahl, R. Chandra, J. Padhye, L. Ravindranath, M. Singh, A. Wolman, B. Zill, Enhancing the security of corporate Wi-Fi networks using DAIR, in: Proc. of ACM MobiSys, June 2006.
[12] R. Chandra, J. Padhye, A. Wolman, B. Zill, A location-based management system for enterprise wireless LANs, in: NSDI, April 2007.
[13] Y.-C. Cheng, M. Afanasyev, P. Verkaik, P. Benko, J. Chiang, A.C. Snoeren, S. Savage, G.M. Voelker, Automating cross-layer diagnosis of enterprise wireless networks, in: Proc. of ACM SIGCOMM, Kyoto, Japan, August 2007.
[14] Y.-C. Cheng, J. Bellardo, P. Benko, A.C. Snoeren, G.M. Voelker, S. Savage, Jigsaw: solving the puzzle of enterprise 802.11 analysis, in: Proc. of ACM SIGCOMM, Pisa, Italy, September 2006.
[15] A. Chhetri, R. Zheng, WiserAnalyzer: A passive monitoring framework for WLANs, in: Proc. of Mobile Ad-hoc and Sensor, Networks, 2009.
[16] U. Deshpande, T. Henderson, D. Kotz, Channel sampling strategies for monitoring wireless networks, in: Proc. of the Second Workshop on Wireless Network Measurements, Boston, MA, April 2006.
[17] U. Deshpande, C. McDonald, D. Kotz, Coordinated sampling to improve the efficiency of wireless network monitoring, in: Proc. of the Fifteenth IEEE International Conference on Networks (ICON), November 2007.
[18] D.S. Hochbaum, Approximating covering and packing problems: set cover, vertex cover, independent set, and related problems, in: D.S. Hochbaum (Ed.), Approximation Algorithms for NP-hard Problems, PWS Publishing Co., Boston, MA, USA, 1996, pp. 94–143.
[19] A.P. Jardosh, K.N. Ramachandran, K.C. Almeroth, Understanding link-layer behavior in highly congested IEEE 802.11b wireless networks, in: Proc. of ACM SIGCOMM Workshop on Experimental Approaches to Wireless Network Design and Analysis (E-WIND), August 2005.
[20] R. Mahajan, M. Rodrig, D. Wetherall, J. Zahorjan, Analyzing the MAC-level behavior of wireless networks in the wild, in: Proc. of ACM SIGCOMM, September 2006.
[21] Y. Sheng, G. Chen, H. Yin, K. Tan, U. Deshpande, B. Vance, D. Kotz, A. Campbell, C. Mcdonald, T. Henderson, J. Wright, MAP: a scalable monitoring system for dependable 802.11 wireless networks, IEEE Wireless Commun. 15 (5) (2008).
[22] Y. Sheng, K. Tan, G. Chen, D. Kotz, A. Campbell, Detecting 802.11 MAC layer spoofing using received signal strength, in: Proc. of IEEE INFOCOM, April 2008.
[23] A. Sheth, C. Doerr, D. Grunwald, R. Han, D.C. Sicker, MOJO: A distributed physical layer anomaly detection system for 802.11 WLANs, in: Proc. of ACM MobiSys, June 2006.
[24] D.-H. Shin, S. Bagchi, Optimal monitoring in multi-channel multi-radio wireless mesh networks, in: Proc. of ACM MobiHoc, 2011.
[25] D.-H. Shin, S. Bagchi, C.-C. Wang, Distributed online channel assignment toward optimal monitoring in multi-channel wireless networks, in: Proc. of IEEE INFOCOM, Mini Conference, 2012.
[26] R. Shravan, S. Vivek, B. Suman, and C. Ranveer. Fluid: improving throughputs in enterprise wireless lans through flexible channelization, in: Proc. of ACM MobiCom, 2011.
[27] Y. Song, X. Chen, Y.-A. Kim, B. Wang, G. Chen, Sniffer channel selection for monitoring wireless LANs, in: Proc. of Wireless Algorithms, Systems, and Applications (WASA), August 2009.
[28] W. Wei, K. Suh, B. Wang, Y. Gu, J. Kurose, D. Towsley, Passive online rogue access point detection using sequential hypothesis testing with TCP ACK-pairs, in: Proc. of ACM SIGCOMM Internet Measurement Conference (IMC), October 2007.
[29] B. Yan, G. Chen, Model-based fault diagnosis for ieee 802.11 wireless LANs, in: Proc. of the International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous), March 2009.
[30] B. Yan, G. Chen, H. Wang, H. Yin, Robust detection of unauthorized wireless access points, Mobile Netw. Appl. (MONET) 14 (4) (2009).
[31] J. Yeo, M. Youssef, A. Agrawala, A framework for wireless LAN monitoring and its applications, in: Proc. of ACM Workshop on Wireless Security (WiSe), October 2004.
[32] J. Yeo, M. Youssef, T. Henderson, A. Agrawala, An accurate technique for measuring the wireless side of wireless networks, in: Proc. of USENIX/ACM

Please cite this article in press as: X. Chen et al., Sniffer channel selection for monitoring wireless LANs, Comput. Commun. (2012), http://dx.doi.org/10.1016/j.comcom.2012.06.005