Computer Networks 64 (2014) 261–272

Contents lists available at ScienceDirect

Computer Networks

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

# Self-tuned distributed monitoring of multi-channel wireless networks using Gibbs sampler

Yufei Wang [a], Rong Zheng [a,b,*], Qixin Wang [a]

[a] The Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China
[b] The Department of Computing and Software, McMaster University, Canada

ARTICLE INFO

*Article history:*
Received 5 January 2013
Received in revised form 22 January 2014
Accepted 11 February 2014
Available online 19 February 2014

*Keywords:*
Wireless side network monitoring
Distributed sniffer
Gibbs sampler
Channel assignment

ABSTRACT

Wireless side monitoring employing distributed sniffers has been shown to complement wired side monitoring using Simple Network Management Protocol (SNMP) and base station logs, since it reveals detailed PHY and MAC behaviors, as well as timing information. Due to hardware limitations, wireless sniffers typically can only collect information on one channel at a time. Distributed algorithms are desirable to determine the optimal channel allocation of sniffer nodes to maximize the information collected. In this paper, we propose Gibbs sampler based algorithms for robust distributed monitoring of multi-channel wireless networks. Among several variants of the base Gibbs sampling approach, we find that most algorithms suffer from high sensitivity to parameter selection. In contrast, Gibbs sampling using a thermodynamic schedule is self-tuned and can adapt to different network configurations. Simulation studies show that the proposed algorithms can achieve faster convergence rate and have higher chance of reaching global optima than traditional Gibbs sampler algorithm.



## 1. Introduction

Deployment and management of wireless devices and networks are often hampered by the poor visibility of PHY and MAC characteristics, and complex interactions at various layers of the protocol stack both inside a managed network and across multiple administrative domains. Accurate and timely estimates of network conditions and performance characteristics are vital to efficient network operations and beneficial to many applications like network resource management and network diagnosis.

Passive monitoring is a technique where a dedicated set of hardware devices called *sniffers* are used to monitor activities in wireless networks. These devices capture transmissions of wireless devices or activities of interference sources in their vicinity, and store packet level or PHY layer information in trace files, which can be analyzed distributively or at a central location. wireless monitoring [1–6] has been shown to complement wired side monitoring using SNMP and base station logs since it reveals detailed PHY (e.g., signal strength, spectrum density) and MAC behaviors (e.g., collision, retransmissions), as well as timing information (e.g., back off time), which are often essential for wireless diagnosis.

In practice, due to hardware limitations, wireless sniffers typically can only collect information on one channel at a time [6]. Further, since most, if not all infrastructure networks utilize multiple contiguous or non-contiguous channels or bands,[1] an important issue is to determine

* Corresponding author at: The Department of Computing and Software, McMaster University, Canada. Tel.: +1 7137432663.
*E-mail address:* rzheng@mcmaster.ca (R. Zheng).

[1] A channel can be a single frequency band, a code in CDMA systems, or a hopping sequence in frequency hopping systems.

http://dx.doi.org/10.1016/j.comnet.2014.02.014
1389-1286/

which set of frequency bands each sniffer operate on, in order to maximize the total amount of information gathered. This is called the *sniffer-channel assignment* problem or *channel assignment* problem for short. It is a challenging problem since monitoring resource is limited, and thus it is infeasible to monitor all channels at all locations all the time due to hardware limitations.

In [7], we introduce a quality of monitoring (QoM) metric defined by the average number of active users monitored, and investigate the problem of maximizing QoM by judiciously assigning sniffers to channels based on the knowledge of user activities in a multi-channel wireless network. Two capture models are considered. The first one, called the *user-centric model* assumes frame-level capturing capability of sniffers such that the activities of different users can be distinguished. The second one, called the *sniffer-centric model* only utilizes binary channel information (active or not) at a sniffer. For the user-centric model, we show that the implied optimization problem is NP-hard, but a constant approximation ratio, can be attained via polynomial complexity algorithms. When the network size is large, it is desirable to have a distributed solution to the sniffer channel assignment problem utilizing only local information.

Annealed Gibbs sampler is a sampling method for stochastic search [8]. It is a special form of simulated annealing, where only local information is exchanged in each step. Annealed Gibbs sampler was first applied to solve distributed power control, channel assignment and user association problems by Kauffmann et al. in [8]. In this paper, we propose Gibbs sampler based distributed algorithms for the sniffer-channel assignment problem under the user-centric model. We consider several techniques to accelerate the speed of convergence and improve the optimality of the baseline scheme, including parallel execution, distorted objective function, and thermodynamic cooling schedule. Among these variants, we find that most algorithms suffer from high sensitivity to parameter selection. Furthermore, there is little guideline as to how to tune parameters such as the initial temperature and cooling factor given by specific objective functions. In contrast, Gibbs sampling using thermodynamic schedule is self-tuned and can adapt to different network configurations. Simulation results using both synthetic traces and real-world measurements demonstrate the superior performance of these variants.

The rest of the paper is organized as follows. In Section 2, we give an overview of existing work on wireless side monitoring and Gibbs sampler-based algorithms for wireless resource management. The problem formulation is presented in Section 3. The annealed Gibbs sampler, the proposed distributed algorithm, and several of its variants are detailed in Section 4. We evaluate their performance in Section 5 through extensive simulation and conclude the paper in Section 6.

## 2. Related work

Wireless side monitoring (a.k.a. wireless monitoring) is an active area of research that has received much attention from several perspectives. There has been much work done on wireless monitoring from a *system-level* approach, in an attempt to design complete systems, and address the interactions among the components of such systems [9,10,1–4]. The authors have argued both qualitatively and quantitatively the need for wireless side monitoring. The work in [9,10] uses AP, SNMP logs and wired side traces to analyze WiFi traffic characteristics. Passive monitoring using multiple sniffers was first introduced by Yeo et al. in [1,2], where the authors articulated the advantages and challenges posed by passive measurement techniques, and discussed a system for performing wireless monitoring with the help of multiple sniffers, synchronization and merging of the traces via broadcast beacon messages. The results obtained for these systems are mostly experimental. Rodrig et al. in [3] used sniffers to capture wireless data, and analyzed the performance characteristics of an 802.11 WiFi network. One key contribution was the introduction of a finite state machine to infer missing frames. The Jigsaw system, which was proposed in [4], focused on large scale monitoring using over 150 sniffers.

The optimal allocation of monitoring resources for maximizing captured information is well studied. In literature, the optimal monitoring is formulated as maximum coverage problem with group budget constraints, which was previously studied by Chekuri and Kumar in [11]. Shin and Bagchi consider the selection of monitoring nodes and their associated channels for monitoring wireless mesh networks [12]. Shin et al. proposed the solutions for optimal monitoring of a multi-channel network, including a greedy approach and two LP rounding approaches [13]. A similar LP rounding approach is also proposed by Chhetri et al. [7], where only channel assignment to the sniffers is considered. All these approaches in [12,14,13,7] are centralized and achieve sub-optimal results.

In contrast, in this paper (and [15]), we propose a fully-distributed solution based on Gibbs sampler for the channel allocation problem. Annealed Gibbs sampler with logarithmic cooling schedule was proven to achieve optimality (at the cost of slow convergence). Empirically, we have shown in this paper that the annealed Gibbs sampler with properly tuned cooling schedule and self-tuned thermodynamic cooling schedule can in fact converge to optimal solutions in tens of iterations. Another distributed algorithm is proposed by Shin et al. [16], which guarantees an approximation ratio of $(1 - \frac{1}{e})$ and has polynomial time complexity. In comparison, our algorithm can guarantee optimality, but may not converge in polynomial time (in the worst case).

The proposed algorithm runs only once during the bootstrapping phase of the network. However, in practice, as the network condition changes, the sniffer-channel assignment needs to be adjusted online. One simple solution is to periodically run our algorithm, so as to reallocate the monitoring resource. Alternatively, we can adopt the mechanism proposed in [16] to trigger the execution of the proposed algorithm when the amount of information falls below a predefined threshold. The work by Arora et al. [14] considers the sequential learning of network parameters together with the channel assignment decisions. Here, we assume the network parameters have been learned and provided as input to the algorithm.