AirLab: Distributed Infrastructure for Wireless Measurements

Vinod Kone, Mariya Zheleva, Mike Wittie, Zengbin Zhang, Xiaohan Zhao,
Ben Y. Zhao, Elizabeth M. Belding, Haitao Zheng, Kevin C. Almeroth
Department of Computer Science, University of California, Santa Barbara

I. Introduction

The importance of experimental research in the field of wireless networks is well understood. So far researchers have either built their own testbeds or accessed third-party controlled testbeds (http://orbit-lab.org) or used publicly available traces (http://crawdad.cs.dartmouth.edu) for evaluation. While immensely useful, all these approaches have their drawbacks. While building own test beds requires cost and effort, third-party controlled test beds do not replicate real network deployments. On the other hand, the publicly available traces are often collected using different software and hardware platforms, making it very difficult to compare results across traces. As a result, observations are often inconsistent across different networks, leading researchers to draw potentially conflicting conclusions across their own studies.

To facilitate meaningful analysis of wireless networks and protocols, we need a way to collect measurement traces across a wide variety of network deployments, all using a *consistent* set of measurement metrics. Widespread multi-faceted data collection will provide multiple viewpoints of the same network, enabling deeper understanding of both self and exterior interference properties, spectrum usage, network usage, and a wide variety of other factors. Furthermore, data collected in this manner across a variety of heterogeneous network types, such as university, corporate, and home environments, will facilitate cross-comparison of observed network phenomena within each of these settings.

II. Architecture And Design

To address the critical need for comparable and consistent wireless traces, we propose AirLab, a publicly accessible distributed infrastructure for wireless measurements. Figure 1 depicts the high level idea of AirLab's architecture. To produce consistent and comparable wireless traces, each node has the same hardware and software stacks (Figure 2). Each node is a mini-form-factor PC with multiple 802.11 a/b/g radios running Linux. We plan to deploy these nodes in research labs, academic institutions and private homes, with each site hosting at-least one AirLab node. Instead of remote login access, each user submits measurement scripts to the central server. Having a centralized control has several key advantages 1) allows to co-schedule compatible measurement tasks together on the same nodes to maximize node utilization 2) each measurement task is allocated its minimum required time slice and 3) greatly simplifies security and improves stability and availability.

One of the major contributions of AirLab's software platform is the Core Measurement API (CAPI), which enables AirLab users to compose their measurement scripts. The guiding principles behind CAPI design are: **flexibility** to allow further extension of its functionality, **consistency** to enable collection of compatible metrics and **efficiency** to interact directly with the radio hardware and let users gather desired metrics on the fly. Users are given the flexibility to write measurement scripts in their favorite scripting language. We provide the tools to interface the measurement scrips with CAPI.

CAPI performs two types of measurements: (i) baseline and (ii) user-defined. Baseline measurements gather continuous traces in the background, whereas user-defined are triggered by users' scripts. Triggers can be time based or event based (e.g. start when channel load $>$ 2 Mbps). We envision the baseline measurements as *passive* and user defined as either *passive* or *active*. While performing *passive* measurements an AirLab node promiscuously monitors a given channel and reports requested metrics. For *active* measurements, CAPI interfaces with traffic generation tools like *iperf*.

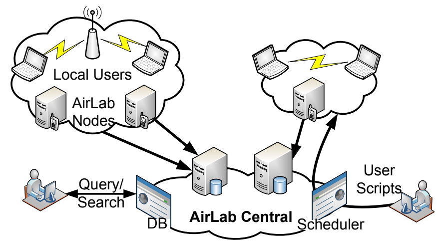

Fig. 1: AirLab Architecture. AirLab includes measurement nodes located at member sites that perform local measurements and send anonymized measurement results back to AirLab Central. Users can query the collected data, as well as schedule customized measurement tasks written as scripts on the AirLab API.

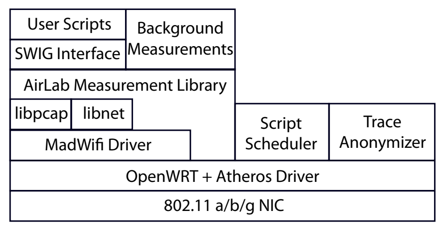

Fig. 2: AirLab Software Stack. Each node has a fixed measurement API on top of MadWifi Atheros drivers. A scheduler periodically informs AirLab Central of local network conditions and receives user scripts. User tasks are executed alongside AirLab's periodic measurements.

Challenges. Our first challenge is understanding and designing the collection of desired metrics. It is crucial for our system to collect metrics valuable for wireless researchers and also be adaptive as requirements evolve in future. Second, and more important, is providing security and data privacy. Through active measurements or compromised user accounts, users can perform (un)intentional DoS attacks on the host wireless networks. To prevent this, we plan to perform both static and run-time checks over the users' scripts to keep traffic rates in check. Also, since AirLab supports collection of packet traces promiscuously, it is possible for users to sniff sensitive information like IP/MAC addresses, open ports etc of the host wireless network. Therefore, it is critical to anonymize the traces to provide privacy, but still not compromise on the functionality of the traces. This is an open and challenging research problem. We plan to leverage the poster session to talk to the experts in the community and get feedback about their requirements from a measurement system and their privacy concerns.