Comparing Wired-side and Wireless-side WLAN Monitoring Techniques: A Case Study

Aniket Mahanti¶, Carey Williamson¶, Martin Arlitt¶§, and Anirban Mahanti†
¶Department of Computer Science, University of Calgary, Canada
§Enterprise Systems and Software Lab, HP Labs, U.S.A.
†Department of Computer Science and Engineering, Indian Institute of Technology Delhi, India

Abstract

Wireless Local Area Networks (WLANs) have become omnipresent: WLANs are available at airports, coffee shops, university campuses, corporate environments, and homes. This surge in the popularity of WLANs motivates the study of how these networks are used. Characterizing WLANs, however, is complicated by a number of factors including the geographic diversity of WLAN deployments and the need for capturing activity in the wireless environment instead of the wired environment. In this paper, we describe our experiences with the deployment and use of a remote passive wireless-side measurement infrastructure for monitoring usage of WLANs, and compare our results with a commonly used wired-side measurement technique.

1 Introduction

The popularity of Wireless Local Area Networks (WLANs) motivates the study of how these networks are used. Today large-scale WLAN deployments can be found in many corporate environments and large universities. Traffic characterization of large WLANs can help network designers understand how users are utilizing the WLAN, which in turn can aid in future service expansions and upgrades. Knowledge of WLAN network usage can also be useful for managing the network. For example, certain applications such as Peer-to-Peer (P2P) are known to be bandwidth-intensive. Excessive use of these applications may result in congestion in the WLAN; thus, a WLAN network administrator may decide to limit bandwidth consumption or block undesired traffic.

The primary challenges for WLAN measurement include the geographic diversity of WLAN deployments, the physical proximity required for WLAN packet capture, and the need for a wireless-side (rather than just wired-side) view of the network. Furthermore, the increasing complexity of deployed WLANs (e.g., multi-channel IEEE 802.11a/b/g networks), the heterogeneity of user equipment (e.g., different devices, operating systems, and protocol stacks), and recent trends in Internet usage (e.g., gaming, P2P file sharing, video streaming) make network traffic measurement challenging.

In this paper we describe our experiences with deployment of a remote passive measurement infrastructure for characterizing our campus WLAN. Our measurement infrastructure uses commercially-available monitoring devices called Radio Frequency Grabbers (RFGrabbers) [19] to collect wireless traffic from 9 selected locations in 7 buildings on the campus WLAN. We compared results obtained from analyzing data from the wired-side (syslog traces) and wireless-side (wireless packet capture) of the WLAN. We did not utilize Simple Network Management Protocol (SNMP) polling of the WLAN as such a technique has been shown to be unreliable [5]. Furthermore, SNMP polling is typically done at intervals of minutes, which makes it unsuitable for comparison with wireless-side traces.

The paper makes three primary contributions. First, our work is of practical importance. We present a methodology and describe the challenges with wireless-side packet capture using remote monitoring devices. Researchers can benefit from our experiences and recommendations when dealing with the measurement challenges. Second, we provide a detailed description of the Aruba syslog messages to understand network activity. Syslog messages are vendor-specific, and often difficult to decipher. Finally, we compare wireless-side monitoring with syslog-based wired-side monitoring mechanisms and show which method is suitable for what types of WLAN analysis. To the best of our knowledge this is the first study to compare the capabilities of syslogs with wireless packet traces.

The remainder of the paper is organized as follows. Our wireless-side trace collection infrastructure is discussed in Section 2. The challenges encountered while deploying the infrastructure are outlined in Section 3. The methodology adopted for comparing results obtained using wireless-side and wired-side monitoring techniques are described in Section 4. Results and discussions are presented in Section 5 and Section 6, respectively. Related work is presented in Section 7. Conclusions are presented in Section 8.

2 Wireless-side Trace Collection

We used off-the-shelf WLAN packet sniffers called RFGrabbers in conjunction with a specialized trace capture program called Airopeek [20] to collect traces from the University of Calgary campus WLAN. The remainder of this section discusses the measurement infrastructure used to obtain traces from the campus WLAN.

2.1 The Airopeek Analyzer

Airopeek is a real-time 802.11 a/b/g WLAN analyzer used for performing site surveys, security audits, application-layer protocol identification, and troubleshooting. Airopeek works in conjunction with a WLAN adapter to "sniff" packets from the air. Airopeek can capture the MAC-layer and the higher-layer protocol headers of a packet.

Two specific features of Airopeek make it attractive for WLAN data capture. First, Airopeek allows multiple simultaneous capture sessions, each using a different adapter. This means that a single workstation with multiple network interfaces can be used to capture multiple concurrent traces, thus reducing the hardware required for the task. Second, for WLANs that operate on multiple channels, Airopeek supports channel scanning on the network adapter. Airopeek can be used for setting the channels that need to be scanned, the order in which they are scanned, and the duration of each scan.

2.2 RFGrabber: The Wireless Packet Sniffer

The RFGrabber probe is an Ethernet-connected WLAN adapter that acts as a "listen-only" AP. With an RFGrabber, one can capture 802.11a/b/g WLAN packets at a remote location and send copies of those packets back to Airopeek running on any network-accessible computer. The captured packets are encapsulated in UDP.

As with most other wireless adapters, RFGrabbers do not guarantee that all packets in the air are captured. The obvious reasons are physical and environmental limitations, such as the number of antennae, operating range, signal quality, and traffic intensity. Typically, RFGrabbers have an indoor operating range of up to 80 meters [20].

RFGrabber placement is thus an important issue for successful trace collection. We therefore tried to place the RFGrabbers near an AP or a set of APs. This choice was motivated by the fact that this placement strategy ensured that more packets are captured; being close to an AP, an RFGrabber can capture packets sent from the clients to the AP, as well as all traffic sent by the AP. Sniffer (RFGrabber) placement and measurement loss estimation is a separate problem, which we have addressed in a separate paper [13].

Ideally, one RFGrabber should be placed for every AP. If the WLAN simultaneously operates on multiple channels, then one RFGrabber should be allocated per channel. However, such deployment may not be feasible, owing to budgetary limitations. Thus, a cost-benefit analysis should be conducted before procuring sniffer devices and deploying them. The number of sniffer devices could vary depending on the purpose of the study. Currently, we have the RFGrabbers deployed at 9 locations that are popular with WLAN users. We have found these locations sufficient for our study.

3 Wireless-side Tracing: Challenges

This section discusses the key challenges we encountered when deploying the trace collection infrastructure. We believe that an outline of the pitfalls encountered and the remedial steps taken to address them will be useful to others intending to conduct wireless-side monitoring of WLANs.

3.1 Preliminary Deployment and Tests

The campus wireless measurement project was conceptualized in May 2005. As mentioned earlier, we were interested in using commercially-available tools. After reading the literature of available products, we decided to use Airopeek along with RFGrabber devices.

Before deploying RFGrabbers throughout the campus WLAN, we conducted trials in the computer science (CPSC) department. In June 2005, the CPSC technical staff deployed three RFGrabber probes on the top three floors of the Information and Communications Technology (ICT) building. Deployment of the devices was quick and easy. After connecting the device to an Ethernet port, it automatically received an IP address from the CPSC DHCP server. Each Ethernet port in the CPSC network is assigned a static IP address. This means that every time the RFGrabber probe was restarted (e.g., after a power outage), the device received the same IP address. Note that the workstation running Airopeek (and collecting traces) contacts the RFGrabber using its IP address. Hence, knowledge of the RFGrabber IP address is critical.

Our preliminary tests indicated that trace sizes were voluminous, if packet filtering was not enabled. Therefore, we decided to collect only packets sent and received by the WLAN APs. Also, for each wireless frame we saved (at most) the first 512 bytes. A packet slicing value of 512 bytes is, for most cases, sufficient to guarantee that all headers and part of the payload are captured. Part of the payload is necessary if application-layer protocol analysis is of interest.

3.2 Trace Collection

The task of learning Airopeek functionality started immediately after the initial deployment. At that time, version 2.0.2 of Airopeek was used.

Airopeek 2.0.2 supports two different trace formats: a generic ASCII Comma Separated Values (CSV) format, and a proprietary format (APC) developed by WildPackets (the manufacturer of Airopeek and RFGrabber). A separate utility provided by WildPackets can convert APC files to `pcap` format, which can be read by common tools like `tcpdump`. Saving the large trace files first in APC format and then using another program to convert them to `pcap` format was deemed unsuitable because of the time required by the conversion process. As a compromise, the CSV format was chosen. The trace files were to be processed using custom-written `Perl` scripts.

Airopeek 3.0.1 became available in November 2005. This improved version was more reliable, and addressed many of the deficiencies of the previous version. A quick switch was made to this version. However, this version did not support saving trace files in the CSV format. Reverting back to version 2.0.2 was unacceptable. Following consultation with WildPackets technical support, we were given a set of `C++` files called PeekRdr that were capable of reading the APC format. PeekRdr was converted to `Perl` and made to work with the existing `Perl`-based analysis script.

3.3 Final Deployment

Deployment across the campus WLAN required assistance from University of Calgary Information Technologies (UCIT). This group manages our campus network. In September 2005, we only had 4 version 1.1 RFGrabbers. Because version 1.1 required a normal AC outlet (usually situated near floor-level) and most APs were mounted on the ceiling or walls, UCIT encountered major difficulties in deploying the RFGrabbers near a set of APs. The problem was amplified by the need for secure locations to prevent theft or vandalism. The first phase of deployment was complete in the beginning of October.

By October 2005, the IP addresses of all RFGrabbers were provided to us. However, when the IP addresses were entered in Airopeek, it was unable to discover the probes. Initially we suspected faulty cabling to be the source of this problem; investigations revealed that this was not the case. Note that our trace collection workstations were situated in the ICT building; the network in this building was administered by CPSC. Troubleshooting showed that the problem occurred because the CPSC network restricts certain traffic such as SNMP from entering into its domain. Also, the CPSC network blocked some UDP ports, including those used by the RFGrabber, namely 161, 37008, and 44033. Following a request to the CPSC technical support staff, port blocking was relaxed for our workstations. Airopeek was then able to connect to the probes.

In December 2005, we purchased 5 new RFGrabbers (version 2.0) to expand the number of locations for collecting traffic. Because the new version supported Power over Ethernet (PoE) [8], we thought that the devices could be placed in any desirable location. Accordingly, 5 new locations were identified. We anticipated that the new PoE RFGrabbers could be placed closer to the AP. To our surprise, we were informed that each AP has special dedicated PoE cabling, which would not be available for the probes. Considering the time and cost required to install the additional cables, we decided to power the new RFGrabbers using AC sources.

After deployment in the campus WLAN, the RFGrabbers experienced occasional connection outages. Every two weeks or so we would lose connection to the RFGrabber. The reason for this problem was the dynamic IP addressing scheme used in the campus network. Thus, if for any reason the probe was restarted it received a new IP. However, Airopeek still had the old IP address and could not connect to it. To solve this problem, a static IP address was assigned to all the campus RFGrabbers. By the end of February 2006, we had 9 RFGrabbers operational at different locations on campus.

4 Methodology: Wireless-side vs. Wired-side

To compare the results obtained from wired-side and wireless-side analysis, we focused on three commonly used WLAN metrics, namely user count, user session characteristics, and user session activity. In this section we describe the methodology adopted to measure these in the Airopeek trace and syslog data. Although we are using a proprietary wireless trace format, the analysis and results are applicable to other wireless-side monitoring techniques.

Distinguishing Users: In the Airopeek trace, WLAN users were distinguished by the MAC addresses of their Network Interface Cards (NICs). We assumed that each MAC address represented a unique user. We looked at the Address fields of all Data frames in the trace. For Data frames sent from each AP, all unique MAC addresses seen in the Address 2 field of the Data frames sent to the AP were added to the list of users. When To-DS=1, Address 2 represents the MAC address of the transmitter station. Similarly, all unique MAC addresses seen in the Address 1 field were added to a list of unique users. When From-DS=1, Address 1 represents the MAC address of the receiver station.

In the syslog data, whenever we saw the `<NOTI> login <MAC IP name>` notification message, we considered the MAC address to be a legitimate user. Additionally, we used messages for user sessions to count active users during any interval. These messages are discussed next.

Distinguishing Sessions: Users generate sessions. A session lasts from the time a user joins the WLAN until they leave the network. The session duration is defined as the time spent between the user joining and leaving the network.

In the Airopeek trace, we designated a session start point if we noticed an exchange of Authentication frames, Authorization frames, and DHCP packets between the AP and the NIC. A session is terminated when the NIC sends a Disassociation frame to the AP or the AP sends a Deauthentication frame to the NIC due to sustained inactivity. In the absence of Authentication and Association frames, we started a new session whenever a packet from a new user was noticed in the trace. Similarly, in the absence of Disassociation frames, to be able to differentiate between two sessions of the same user, we chose a session timeout of 30 minutes.

In the syslog, we identified the start of a user session when we saw the following message sequence:

```
<NOTI> auth req <MAC> AP
<NOTI> auth success <MAC> AP
<NOTI> assoc req <MAC> AP
<NOTI> assoc success <MAC> AP
```

where `<MAC>` and `AP` represent the MAC addresses of the user NIC and AP, respectively. A session was initiated when we observed the `auth success` message. Compared to the Airopeek trace, here we have more information regarding sessions, and by analyzing these messages we can distinguish between a new session and an existing session, even if session timeout (30 minutes) is not reached. Note that syslog messages are sent encapsulated in UDP.

We observed the following message sequence when a user who had previously shutdown their wireless device starts a new session:

```
<INFO> station up <MAC> bssid AP, essid airuc
<INFO> user vlan <MAC> assigned x, default x,
       current x, bssid=MAC
<INFO> station up <MAC> update station bssid
       to MAC (users 0)
<INFO> user mobility <MAC IP> INTER MOVE: ...
<INFO> user add <MAC IP> mobility: ...
<INFO> inherit <MAC IP> bssid:MAC essid: airuc ...
<INFO> TRAIL: MAC IP MS: IP V: IP ...
```

When a user roamed during a session we observed the following message sequence:

```
<INFO> station down <MAC> bssid AP, essid airuc ...
<INFO> station up <MAC> bssid AP, essid airuc...
<INFO> user vlan <MAC> ...
<INFO> station up <MAC> update station bssid to MAC...
<INFO> user mobility <MAC IP> ...
<INFO> TRAIL: MAC IP MS: IP V: IP ...
```

These messages indicate how session initiation may be detected; if the messages `inherit`, `user miss`, or `user add` are seen, then it indicates that the user established a new session. For example, suppose a user switched off their wireless device and turned it on again a few minutes later. It is likely that the user would not be prompted to reauthenticate again unless the user had explicitly logged out of the WLAN or the session had expired.

During a session, we focus on the user's TRAIL messages in the WLAN, and update the state each time a message regarding the user is seen. Some of messages contained AP information, thus allowing us to detect roaming within a session. The following message sequence describes the preceding scenario:

```
<INFO> TRAIL: MAC IP MS:IP V:# L:#.#.# T:AP #:MAC
       Frame Retry Rate at # for STA MAC AP MAC
<INFO> station down <MAC> bssid MAC, essid airuc
<INFO> station up <MAC> update station bssid to MAC...
<INFO> station up <MAC> bssid MAC, essid airuc ...
```

Some syslog messages did not contain AP information. Nevertheless, these messages indicated that the session was active. As before, we updated the last time a message concerning an active user was seen, and used this updated value to calculate session idle timeout. An example of such a syslog message sequence is:

```
<INFO> DHCP handshake complete for user (MAC IP)
<INFO> Adding bridge entry for MAC
<INFO> Removing bridge entry for MAC
<INFO> user mobility <MAC IP>
<INFO> mob update <MAC IP name>
<INFO> Requesting AUT for MAC
```

We terminated a session when we saw the following message:

```
<NOTI> disassoc from sta <MAC> AP MAC
<NOTI> logout <MAC IP name>
<INFO> user del <MAC IP name>
```

We calculated session duration as the difference of `logout/disassoc/user del` message time and session start time. Hence, we had two session end conditions: (a) inter-message timeout exceeds 30 minutes, or (b) we saw one of the session end messages. Also all sessions shorter than 30 seconds were ignored because these indicated cases when the user associated and disassociated due to error (message: `reason unspecified failure`). Note that all the aforementioned syslog messages contained the MAC address of the user NIC, which was used to keep track of the number of active users during an hour or day.

During peak hours, some session start messages were lost. If this was the case and we saw one of following messages, a new session was started:

```
<INFO> station up <MAC> bssid MAC, essid airuc...
<INFO> station up <MAC> update station bssid to MAC
<INFO> user vlan <MAC> ... bssid=MAC
```

As mentioned above, inherit, user add, and user miss messages indicate a new session start; however, there were cases when we saw such messages soon after station up or any other message indicating a session start. In these cases, if before receiving the inherit message we had started a session within 5 minutes before inherit or user add, then we assumed that the messages were related to an active session and we did not close the session. Here we assigned the session's start time to be the timestamp of inherit or user add message. This is frequently observed during peak hours. Thus to avoid closing already started sessions we used another timeout. If inherit, user add, and user miss were observed within the first 5 minutes of a session, then we only updated session start times. However, if the time difference exceeded 5 minutes, then this meant that the user had switched off the wireless device and had turned it on again. The previous session was closed and a new session was started at this point. The duration of the previous session is equal to the difference between the session start time and the last seen message time before inherit.

5 Results

Our university WLAN (named *AirUC*) consisted of 500 APs at the time of trace capture. Most of the APs in the AirUC network are Aruba 70 dual-band 802.11 a/b/g APs. The Aruba 70 is a dual-radio “thin” AP with built-in omni-directional high-gain tri-band antenna to support the 2.4 GHz and 5 GHz spectrums. Thin APs implement the minimal functionality required by the 802.11 standard. Upper-layer MAC processing functions are integrated into a central AP controller. The AP controller used in the campus is the Aruba 6000. The WLAN employs Web-based authentication using the Aruba captive portal. We configured the RFGrabbers to scan channels 1, 6, and 11 every 500 ms, to observe the channel spectrum used by our WLAN in the ‘b/g’ mode. The RFGrabbers were able to capture packets from 97 APs, representing about 20% of the APs in the WLAN.

This section presents some preliminary analyses of the Airopeek traces collected from the 9 campus locations. These locations represent social areas (e.g., coffee area, food court), academic areas (e.g., classrooms), libraries, and service areas (e.g., IT office) found in the campus. The purpose of these analyses is to gauge the accuracy of the captured traces in understanding user and network activity at those locations. We compare our wireless trace analyses with syslog data collected from the wired-side of the network. Both data sets where collected between April 1, 2006 12:00 AM and April 6, 2006 11:59 PM. We sanitized the syslog trace to contain messages that only related to the APs and users monitored by the RFGrabber probes.

5.1 Number of Users

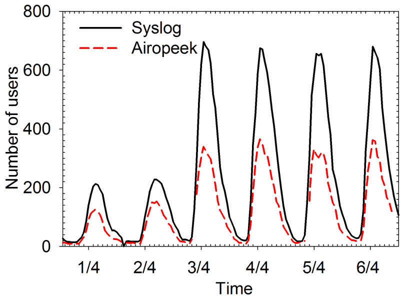


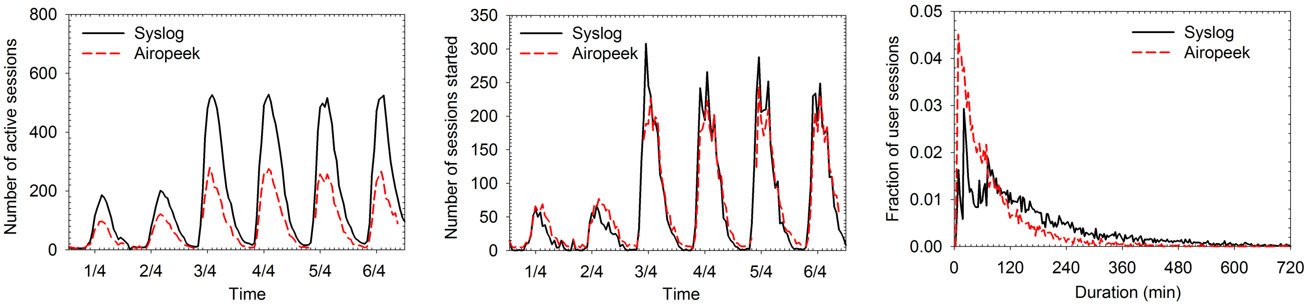
Figure 1. Number of users per hour

Figure 1 compares the number of users seen in an hour using both traces. Usage follows a clear diurnal pattern, with most users accessing the network on weekdays between 10 AM and 1 PM. We found that the number of users calculated using the Airopeek trace was less than that from the syslog trace. Losses at the probes were likely responsible for this phenomenon. In total, 2,955 unique users were identified using Airopeek, whereas 3,456 users were identified using syslog during the 6 day trace duration. We believe that the 501 users not identified by Airopeek were outside the probes' operating range.

5.2 User Sessions

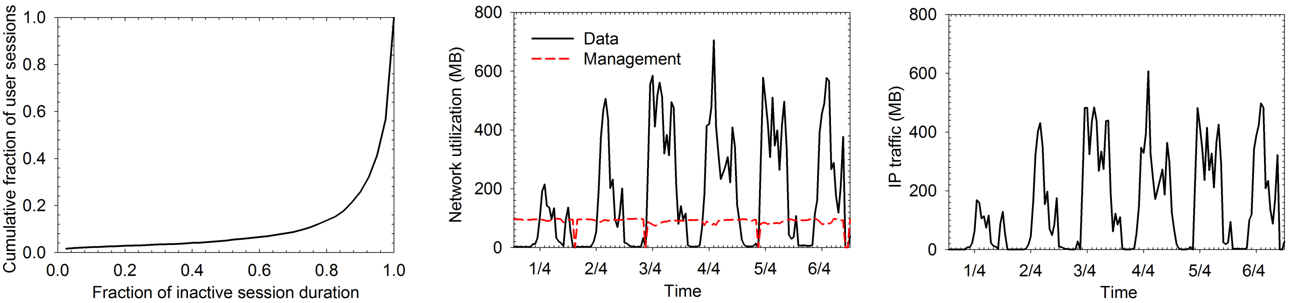
Figure 2(a) shows the number of active sessions per hour. We considered a session active if the idle time (i.e., no data transfer) was less than 20 minutes. We observe that the results in Figure 2(a) match well with the results in Figure 1. We also notice that there are fewer active sessions than active users per hour, indicating many users accessed the WLAN for short periods.

Figure 2(b) shows the number of sessions started in an hour, using Airopeek and syslog traces. The campus WLAN uses a Web authentication system. Hence, for Airopeek traces a session is started when the NIC is authenticated by the AP. Syslog messages report both AP and Web authentication. Both results coincide indicating that



(a) Active sessions per hour (b) Session starts per hour (c) Session duration distribution

Figure 2. User session characteristics



(a) Session inactivity distribution (b) Network utilization per hour (c) IP traffic per hour

Figure 3. User sessions activity

Airopeek traces can identify user sessions adequately. Using Airopeek traces we identified 7,607 unique user sessions. From the syslog data we found 8,393 unique user sessions. Airopeek timestamp granularity is in order of microseconds, while syslog trace granularity is only one second. Thus, Airopeek may be used to more precisely understand the interarrival process of sessions.

Figure 2(c) shows a PDF of session duration using Airopeek and syslog traces. The tails of the durations coincide for both traces. The differences in results are due to the syslog message reporting system. For both methods we employed a session timeout period, as we often did not find the appropriate frame (in case of Airopeek) or message (for syslog) to end a session.

Syslog traces only have session management messages, while Airopeek records all data, control, and management traffic. Using RFGrabbers we continuously monitor WLAN traffic and even a single packet captured before the timeout period indicates that a session is still active. With syslog, infrequent user activity combined with message loss may result in inaccurate session duration estimates. In general, results from Airopeek and syslog may differ; using Airopeek we may estimate long sessions whereas using syslog a session may be split into multiple smaller duration sessions each of varying durations.

5.3 User Session Activity

In this section we demonstrate several analyses that can only be performed using wireless-side packet traces. We used the Airopeek traces for this purpose. When using a wired-side measurement infrastructure, such analyses require supplementary data collection. For example, SNMP polling can be used to understand network load, syslog for user and session activity, and Ethernet traces for application classification.

Figure 3(a) shows the period of inactivity during a session. Because syslog only has session management messages, we cannot determine for what fraction of time a session is idle. The figure shows that approximately 45% of the sessions were inactive for 95% of their duration. We conjecture that users use the network for some period of time (e.g., surf the Web), while reverting back to offline activities the rest of the time.

Using the RFGrabber probes we captured 24 GB of data traffic and 12 GB of management traffic. Figure 3(b) shows network utilization for the trace period. We observe a steady stream of management traffic mostly due to APs transmitting beacons. The data traffic is dependent on users. We observe that network utilization is comparatively lower on April 1 (Saturday) and April 2 (Sunday). There are fewer students on campus during the weekend resulting in lower load on the WLAN. The peaks represents weekdays and the dips indicate nights.

Figure 3(c) shows the amount of IP traffic generated by users during the trace period. These results fit well with the results shown in Figure 3(b) indicating that most (20 GB) but not all data traffic was generated by users accessing the WLAN. A protocol analysis of the trace showed that about 47% of the user traffic was due to Web surfing. About 15% of the traffic was due to P2P applications such as BitTorrent and MP2P. Streaming media traffic, network services, E-mail, and chat programs accounted for less than 15% of the traffic bytes. In terms of volume, about 96% of the IP traffic used TCP, while only 3% used UDP.

6 Discussion

In the preceding section we saw the differences in how wired-side and wireless-side traces measure user and session counts. In this section we analyze the reasons for these differences.

6.1 Differences in Counting Active Sessions

From Section 5.1 and 5.2 we notice that Airopeek and syslog results match for session starts (Figure 2(b)), while the results for active sessions do not match (Figure 2(a)).

The mismatch arises due to problems in precisely determining the end of a session in the syslog trace. Syslog messages only report updates to the state of the user. Thus, if the user is stationary and idle there would be no updates to the syslog until the connection is terminated due to inactivity or the user shuts off the device. We were also limited by the fact that we had filtered the syslog data based on the user and AP MAC addresses found in the Airopeek trace and that not all messages in the syslog contained the MAC addresses. The syslog data had gaps and sometimes messages were out of order, especially during peak hours.

Figure 4 illustrates the differences between session starts (*SS*) and active sessions (*AS*), as calculated using Airopeek and syslog traces. The figure shows two sessions each of 6 hours duration. These sessions span multiple parts of the WLAN, some of which are not monitored by the RFGrabbers. These instances are depicted using broken grey lines on the timeline. Solid lines represent parts of the session that are captured by the RFGrabbers. Figure 4(a) describes the true view of the WLAN used by User1 and User2. Figure 4(b) shows the hourly counts for session starts and active sessions when analyzing the Airopeek trace. Figure 4(c) shows the *SS* and *AS* counts when analyzing syslog with gaps or out-of-order messages. Figure 4(d) shows the *SS* and *AS* counts when analyzing the syslog with no holes or disordered messages.

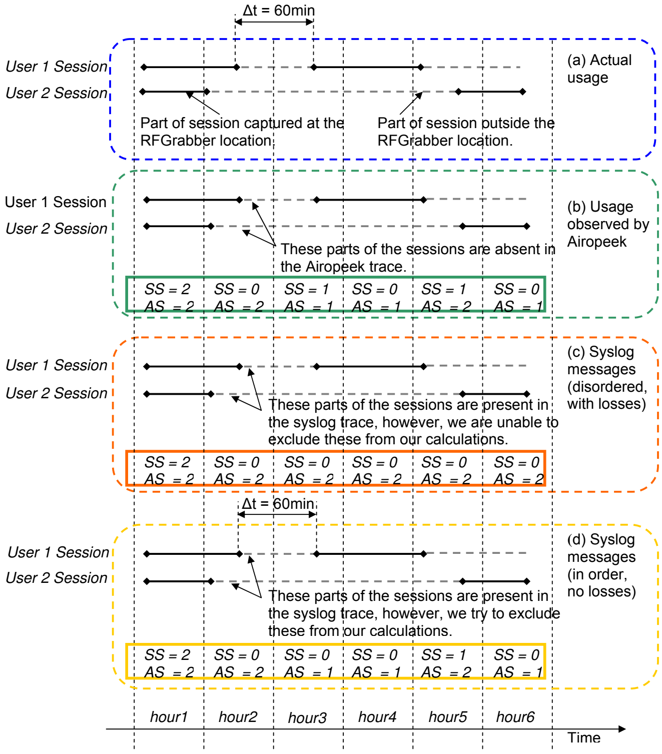

Δt = 60min
User 1 Session
User 2 Session
Part of session captured at the RFGrabber location.
Part of session outside the RFGrabber location.
(a) Actual usage
User 1 Session
User 2 Session
These parts of the sessions are absent in the Airopeek trace.
(b) Usage observed by Airopeek
SS = 2 AS = 2
SS = 0 AS = 2
SS = 1 AS = 1
SS = 0 AS = 1
SS = 1 AS = 2
SS = 0 AS = 1
User 1 Session
User 2 Session
These parts of the sessions are present in the syslog trace, however, we are unable to exclude these from our calculations.
(c) Syslog messages (disordered, with losses)
SS = 2 AS = 2
SS = 0 AS = 2
SS = 0 AS = 2
SS = 0 AS = 2
SS = 0 AS = 2
SS = 0 AS = 2
Δt = 60min
User 1 Session
User 2 Session
These parts of the sessions are present in the syslog trace, however, we try to exclude these from our calculations.
(d) Syslog messages (in order, no losses)
SS = 2 AS = 2
SS = 0 AS = 2
SS = 0 AS = 1
SS = 0 AS = 1
SS = 1 AS = 2
SS = 0 AS = 1
hour1
hour2
hour3
hour4
hour5
hour6
Time

Figure 4. Comparison of session starts and count of active sessions for Airopeek and syslog

In Figure 4(b) we see that in *hour1* two sessions are started and there are two active sessions. In hour2, these sessions continue and hence $SS = 0$. During *hour2* both users move to a location that does not have any probes and hence we see a break in the session as per the Airopeek trace. In *hour3*, *User1* moves to a location monitored by the probes. Because the last packet seen from *User1* was 60 minutes ago (greater than the session timeout), we consider the previous session to be closed and initiate a new session. A similar scenario is noticed in case of *User2* during *hour5*. Hence, during *hour5* $SS = 1$. $AS = 2$ as there are two active sessions during *hour5*.

In syslog, we only considered messages containing the MAC address of a monitored AP. When the user NIC associated with any other AP we ignored the corresponding messages, unless the user returned and associated with a monitored AP. This systematic filtering was essential to synchronize the syslog and Airopeek datasets. However, many syslog messages did not contain MAC addresses of the APs. In such cases we assumed that the user remained associated with the same AP until we received a message stating otherwise. Typically we received the following four messages when a user NIC roamed from one AP to another:

```
1. <INFO> station down <NIC MAC> bssid
   (AP2), essid airuc
2. <INFO> station up <NIC MAC> bssid
   (AP1), essid airuc
3. <INFO> Removing bridge entry for (NIC MAC)
4. <INFO> user mobility <(NIC MAC)>
```

Note that messages 3-4 do not contain information about the AP. Here, we do a reverse lookup to find the AP with which the user NIC was previously associated and make an appropriate decision. So, if `AP1` is monitored by the probes and `AP2` is not, after seeing the second message we will eventually terminate the session if the user did not associate with one of the monitored APs within the session timeout threshold. However, if syslog missed the second message, we continue to (incorrectly) assume that the user is associated with `AP1`. This session will be absent in the Airopeek trace, although it is counted when analyzing the syslog trace. This scenario is depicted in Figure 4(c). When all messages are correctly received, the results match with the Airopeek analysis, as shown in Figure 4(d).

Syslog messages may not arrive in the correct order. For example, the second message could come before the first message. In such a case, analysis of the syslog would indicate that the session is still active, while it would have closed as per the Airopeek trace. This is due to assigning all update records to the last seen AP in the syslog. We have noticed that `station up` messages are often missed in the syslog trace.

These idiosyncrasies of syslog messages cause the active session analysis to differ from that using Airopeek traces. Because sessions are generated by users, user count is also affected. However, as can be seen from Figure 4, session start analysis is not affected.

6.2 Differences in Session Duration Calculation

The reasons for the differences in session durations calculated using syslog and Airopeek traces are explained using Figure 5. Objects in grey with broken borders relate to Airopeek, while items in black with solid borders relate to syslog. The figure shows an example where we collect Airopeek traces from two locations, namely, *Loc2* and *Loc3*. Suppose a user from *Loc1* starts a new session, which is recorded in the syslog. When the user moves to *Loc2*, the RFGrabber probes capture station traffic and a session is started according to the Airopeek trace. Note that we would not see a session start at this timestamp in syslog, as the session had already started earlier in *Loc1*.

When the user moves to *Loc4*, the session timeout threshold is applied to the Airopeek trace, as the station has moved out of the RFGrabbers' range. Thus, the session duration for the user in the Airopeek trace would be the duration that the user was in *Loc2* and *Loc3*. Note that in the syslog trace all messages related to *Loc1* and *Loc4* are ignored as we are only interested in messages for APs that are monitored by the probes. As soon the user roams to *Loc4*, the corresponding syslog update messages will be ignored. If the station returns to *Loc2*/*Loc3* before the session timeout threshold, then the session is allowed to continue. Otherwise, the session is closed and the duration is calculated as the difference in the timestamps of the first seen packet and the last seen packet in *Loc2*/*Loc3*.

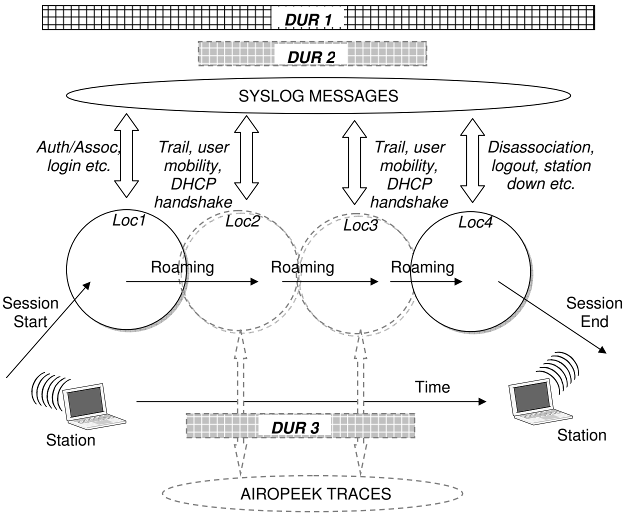


Figure 5. Comparing session durations using Syslog and Airopeek

For this example, the real session duration in syslog is *DUR1*. The session visible for Airopeek is *DUR3*. After filtering all messages related to *Loc1* and *Loc4* in the syslog the approximate session duration is *DUR2*, which matches with the result from Airopeek trace analysis. In syslog we can distinguish between 2 adjacent sessions, but not in the Airopeek traces. Hence, most adjacent sessions get reported as one long session in Airopeek (unless difference between sessions is greater than 30 minutes), while in syslog these are reported as a series of short sessions.

As mentioned earlier, some syslog messages may not contain AP information. For example `<INFO> user add <MAC>`. In such cases, we start the session assuming that user was at *Loc2* or *Loc3*. If we further do not see messages containing AP information from *Loc2* or *Loc3*, then the session is ignored. The 30-minute session idle timeout is well suited for Airopeek trace analysis. However, in some cases this leads to incorrect session termination in the syslog data. In the Airopeek trace we continuously capture traffic and even a single packet captured within 30 minutes indicates that the session is still active. However, in syslog due to message loss and the user being idle, no new update messages can be received within 30 minutes. We consider these sessions to be terminated. In many cases this does not always indicate an end of a session resulting in differences of analysis produced by the Airopeek and syslog traces. When a user actively uses the WLAN the probes capture packets frequently, however, in syslog we only see session management traffic, which depends on user mobility and not session traffic.

7 Related Work

Prior research characterized WLANs on campuses [6, 7, 11, 16, 18], in enterprises [2], and at public hotspots [1, 3, 14]. These measurement studies analyzed data collected from the wired portion of the network. For example, Henderson *et al.* [6] used SNMP polling logs, syslog, and tcpdump, Balazinska and Castro [2] used SNMP polling logs, and Schwab and Bunt [16] used authentication logs and Etherpeek traces.

More recent WLAN studies have used passive wireless-side measurement. Yeo *et al.* [21, 23] addressed the issue of sniffer placement. The authors found that using multiple sniffers can reduce the number of uncaptured frames. They suggested that one sniffer be placed near the target AP, while remaining sniffers be positioned close to the predicted locations of clients. Using this methodology they studied MAC layer characteristics of a department WLAN [22]. Jardosh *et al.* [9, 10] used three laptop sniffers to capture wireless packets from an IETF meeting and studied link-layer behaviour in a congested WLAN. Rodrig *et al.* [15] took wireless measurements using five PC sniffers from the SIGCOMM 2004 conference WLAN to study the operational behaviour of the 802.11 MAC protocol.

Other studies have focused on building wireless-side trace aggregation systems and developed inference mechanisms. Cheng *et al.* [4] developed a system called Jigsaw that provides large scale synchronization of wireless traces from distributed sniffers. Mahajan *et al.* [12] developed a tool (Wit) to merge traces from multiple monitors, infer missed frames, and evaluate WLAN performance. Sheth *et al.* [17] built a system consisting of multiple wireless sniffers, a data collection mechanism, and an inference engine to detect anomalies at the physical layer.

Our work is mostly orthogonal to these prior works. Specifically, we focus on the deployment challenges of a large passive distributed trace collection infrastructure. We found that the number of sniffers deployed and their placement depends on the type of analysis required. A cost benefit analysis is also essential to ascertain the economics of the deployment and the scalability of the system. We compared measurements from wireless-side and wired-side traces, and discussed the underlying reasons for discrepancies in the resulting analysis. Researchers and practitioners can use this information to decide the most appropriate WLAN monitoring mechanism for their needs.

8 Conclusions

This paper describes the challenges we faced during the deployment and subsequent use of a remote, passive, wireless-side measurement infrastructure for monitoring a campus WLAN. We discovered that deployment of wireless packet sniffers across a geographically-distributed WLAN requires cooperation among different groups that manage the network. We also found that network administration policies can impact the measurements, and our experience suggests that some of the purported advantages of wireless packet sniffers may be offset by deployment challenges. We also observed that software and hardware updates are sometimes not compatible with already-deployed technology. We concluded the paper with a comparison of wireless-side and wired-side measurements.

9 Acknowledgements

Financial support for this research was provided by Informatics Circle of Research Excellence (*i*CORE) in the Province of Alberta, as well as by Canada's Natural Sciences and Engineering Research Council (NSERC). The authors thank David Yang, Michael Martini, and Terry Bellward of the University of Calgary Information Technologies Division for deploying the RFGrabber probes in the campus and providing us with the syslog data. The authors also thank the anonymous reviewers for their constructive comments on an earlier version of this paper.

References

[1] A. Balachandran, G. Voelker, P. Bahl, and V. Rangan. Characterizing User Behavior and Network Performance in a Public Wireless LAN. In *Proc. of ACM SIGMETRICS Conference*, pages 195–205, Marina del Rey, U.S.A., June 2002.

[2] M. Balazinska and P. Castro. Characterizing Mobility and Network Usage in a Corporate Wireless Local-area Network. In *Proc. of Conference on Mobile Systems, Applications and Services (MobiSys)*, pages 303–316, San Francisco, U.S.A., May 2003.

[3] D. Blinn, T. Henderson, and D. Kotz. Analysis of a Wi-fi Hotspot Network. In *Proc. of Workshop on Wireless Traffic Measurements and Modeling (WiTMeMo)*, pages 1–6, Seattle, U.S.A., June 2005.

[4] Y. Cheng, J. Bellardo, P. Benko, A. Snoeren, G. Voelker, and S. Savage. Jigsaw: Solving the Puzzle of Enterprise 802.11 Analysis. In *Proc. of ACM SIGCOMM Conference*, pages 39–50, Pisa, Italy, September 2006.

[5] T. Henderson and D. Kotz. Problems with the Dartmouth Wireless SNMP Data Collection. Tech. Report, Dartmouth College, 2003. http://www.cs.dartmouth.edu/~dfk/papers/henderson:problems.pdf.

[6] T. Henderson, D. Kotz, and I. Abyzov. The Changing Usage of a Mature Campus-wide Wireless Network. In *Proc. of Conference on Mobile Computing and Networking (MobiCom)*, pages 187–201, Philadelphia, U.S.A, September 2004.

[7] R. Hutchins and E. Zegura. Measurements from a Campus Wireless Network. In *Proc. of IEEE International Conference on Communications (ICC)*, pages 3161–3167, New York, U.S.A., April 2002.

[8] IEEE 802.3af. Power over Ethernet Standard. http://grouper.ieee.org/groups/802/3/af/.

[9] A. Jardosh, K. Ramachandran, K. Almeroth, and E. Belding-Royer. Understanding Congestion in IEEE 802.11b Wireless Networks. In *Proc. of Internet Measurement Conference (IMC)*, pages 279–292, Berkeley, U.S.A, October 2005.

[10] A. Jardosh, K. Ramachandran, K. Almeroth, and E. Belding-Royer. 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)*, pages 11–16, Philadelphia, U.S.A., August 2005.

[11] D. Kotz and K. Essien. Analysis of a Campus-wide Wireless Network. In *Proc. of Conference on Mobile Computing and Networking (MobiCom)*, pages 107–118, Atlanta, U.S.A., October 2002.

[12] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Analyzing the MAC-level Behavior of Wireless Networks in the Wild. In *Proc. of ACM SIGCOMM Conference*, pages 75–86, Pisa, Italy, September 2006.

[13] A. Mahanti, M. Arlitt, and C. Williamson. Assessing the Completeness of Wireless-side Tracing Mechanisms. In *Proc. of IEEE Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Helsinki, Finland, June 2007.

[14] T. Ojala, T. Hakanen, T. Mäkinen, and V. Rivinoja. Usage Analysis of a Large Public Wireless LAN. In *Proc. of Conference on Wireless Networks, Communications, and Mobile Computing (WirelessCom)*, pages 661–667, Maui, U.S.A., June 2005.

[15] M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, and J. Zahorjan. Measurement-based Characterization of 802.11 in a Hotspot Setting. In *Proc. of ACM SIGCOMM Workshop on Experimental Approaches to Wireless Network Design and Analysis (E-WIND)*, pages 5–10, Philadelphia, U.S.A., August 2005.

[16] D. Schwab and R. Bunt. Characterising the Use of a Campus Wireless Network. In *Proc. of IEEE INFOCOM Conference*, pages 862–870, Hong Kong, China, March 2004.

[17] A. Sheth, C. Doerr, D. Grunwald, R. Han, and D. Sicker. MOJO: A Distributed Physical Layer Anomaly Detection System for 802.11 WLANs. In *Proc. of Conference on Mobile Systems, Applications and Services (MobiSys)*, pages 191–204, Uppsala, Sweden, June 2006.

[18] D. Tang and M. Baker. Analysis of a Local-area Wireless Network. In *Proc. of Conference on Mobile Computing and Networking (MobiCom)*, pages 1–10, Boston, U.S.A., August 2000.

[19] WildPackets. http://www.wildpackets.com/.

[20] WildPackets. *Airopeek NX User Manual*. 2003.

[21] J. Yeo, M. Youssef, and A. Agrawala. A Framework for Wireless LAN Monitoring and its Applications. In *Proc. of ACM Workshop on Wireless Security (WiSe)*, pages 70–79, Philadelphia, U.S.A., October 2004.

[22] J. Yeo, M. Youssef, and A. Agrawala. Characterizing the IEEE 802.11 Traffic: Wireless Side. Technical Report CS-TR 4570, Department of Computer Science, University of Maryland, March 2004. http://www.cs.umd.edu/~moustafa/papers/CS-TR-4570.pdf.

[23] J. Yeo, M. Youssef, T. Henderson, and A. Agrawala. An Accurate Technique for Measuring the Wireless Side of Wireless Networks. In *Proc. of Workshop on Wireless Traffic Measurements and Modeling (WiTMeMo)*, pages 13–18, Seattle, U.S.A., June 2005.