Demo Abstract: Mobility in a large-scale WiFi network - From syslog events to mobile user sessions

Jennie Steshenko, Vasanta G. Chaganti, James Kurose
University of Massachusetts, Amherst
140 Governors Dr., Amherst, MA, 01003
{jenstesh,vchaganti,kurose}@cs.umass.edu

Categories and Subject Descriptors
C.2.1 [Network Architecture and Design]: Wireless Communication; C.2.3 [Network Operations]: Network monitoring

Keywords
User Mobility, 802.11 WiFi Networks, Measurements, Syslog, ARUBA

1. INTRODUCTION

We live in a time where our phones, computers, and other devices leave a digital footprint behind us, that includes our network and physical locations. This data can be useful to the research community for many purposes, including the evaluation of mobility architectures and protocols in “real-world” scenarios [4, 5, 7, 10, 11].

This paper describes an ongoing project that uses network management logs from a campus 802.11 wireless network of nearly 4,500 ARUBA access points at the University of Massachusetts Amherst (UMass), to characterize the mobility of tens of thousands of network users. Our own particular use of these traces are to investigate (i) the use of differential privacy in publishing anonymized mobility traces, (ii) the level of complexity of queuing networks needed to accurately model network and user-level performance in mobile networks, and (iii) the performance of new mobility architectures and protocols such as those in MobilityFirst [1] using trace-based, measured mobility in real networks.

We present the processing steps that transform a log of individual network events (primarily AP (dis)association and (de)authentication events) into user session trajectories among APs around the UMass campus, with user transitions among physical locations inferred from these events.

We describe challenges involved in transforming these individual events into mobile user sessions and we overview user mobility characteristics evidenced in these traces. We have shared these traces with several research groups both inside and outside of UMass.

2. NETWORK EVENT DATA

The UMass network consists of approximately 4,500 APs, and is typically used by approximately 35,000 students and staff. The APs are managed by 10 to 12 controllers, each of which controls up to 512 APs. The controllers log network management and user (dis)association from/to AP events (SNMP) and network (de)authentication events (DHCP) [2] to a centralized log file. The (dis)association and (de)authentication events are logged separately per event, per user. Each log file covers one calendar day from midnight to the following midnight. The clocks are synchronized by the ARUBA controllers using a Network Time Protocol (NTP) that provides synchronization to within 1ms interval, providing sufficient granularity for our purposes. Currently, we have access to 90 log files, each containing between 2 GB and 4 GB of data.

We extract raw event data from these logs and transform them into per-user movement trajectories among APs. For detailed information about the structure of ARUBA syslog messages, refer to [13]. A typical message consist of the following parts (each part is enclosed within question marks):


In the sample message below, a disassociation message -<501102>, has been logged by the station (user) with MAC address 9c:e6:35:f8:c3:cb. The message body indicates that the user has disassociated and has left the AP with name GRGH-309-1 and BSSID 00:24:6c:bf:78:al:

Jan 20 20:00:01 lgrc-wac-106-4 stm[2014]:
3. DATA PROCESSING
The individual syslog events are transformed into movement trajectories with these high level steps: (i) processing the syslog data into (start, end time) presences associated with access points; and (ii) agglomerating presences into sessions. In the future we plan to investigate ping-pong effects, and the thresholds for removing ping-pons from the data.

In our processing, Python is used to implement regular expressions parsing over the syslog files to identify message types, as well as the 802.11 finite state machine (FSM) [3, Section 10.3] to extract presences at each AP; MongoDB is used to store the data through the intermediate steps, and to store the final user trajectories. Both Python and MongoDB offer easy out-of-the-box parallelizable packages, improving performance with such large datasets.

The steps to process the raw syslog data files are given in Procedure 1. First we scan through the log files, parsing message parts using regular expressions. If the message_subtype describes user (dis)associations and (de)authentication (to/from the network), the message_body is parsed with an additional set of regular expressions (an example is not given as each subtype has a different structure, and we process a list of subtypes). The extracted tuples are aggregated per MAC address and ordered by the time stamp. This collection is saved to MongoDB.

The aggregated collection is then passed to the FSM to obtain a sequence of “presences” at APs. A presence corresponds to state 4 of the FSM (the user is associated and authenticated); each user first establishes an association with an AP followed by an authentication via the four-way handshake. When a user either deauthenticates or disassociates, the presence at that AP ends. The presences are ordered by the starting time and saved as a new collection in MongoDB. For a given MAC address, each entry contains the start time and end times of the presence at an AP, the name of the AP, and the BSSID.

For the sample message above, assuming the MAC had a previous association message with the timestamp Jan 20 19:00:01.1, the following is a possible entry in a trajectory:

Jan 20 19:00:01, Jan 20 20:00:01,ORGH-309-1,00:24:6c:bf:78:al}$

As the final processing step, we break a MAC’s sequence of presences into user sessions. We define a session as an ordered sequence of presences separated by a gap of time, for which no events are recorded, less than gap length, $G$. To select the the gap length $G$ for which no events are recorded, less than gap length, $G$

$G \leq 1$ minute. A large fraction of hold times, $H(t) \leq 1$ minute, are due to multiple (dis)association events to the same AP, or neighboring APs (ping-pong effects). A second peak observed at $H(t) \approx 15$ minutes corresponds to the idle-timeout of a user, if the device does not respond to an ARP (Address Resolution Protocol) request which is set by default within ARUBA. The subsequent peaks observed with $H(\Delta t) \approx 15$ minutes, occur when the user’s device responds to the ARP request just before the idle time out. The mean hold time at an AP was observed to be 12 minutes.

Each session of a user’s trajectory consists of a sequence of presences at several APs. Within each session, however, we also observe consecutive presences to the same AP. If the gap length between two consecutive presences to the same AP is less than an aggregation gap length $G_{agg}$; we aggregate the presences. The results of aggregating successive presences at the same AP are shown in Fig. 5. The figure shows the mean hop count for user trajectories as a function of the aggregation gap. As $G_{agg}$ increases, there is a decrease in the mean number of hops per session. From the

**Procedure 1** Processing Steps to transform syslog messages to presences.

1: while Scan raw text file do
2: Parse syslog message $\rightarrow$ regular expression
3: if message subtype $\in \{\text{de}auth, \text{(dis)assoc}\}$ then
4: Parse message body $\rightarrow$ regular expression
5: Extract Tuple: (timestamp, message subtype, user (MAC, IP), AP (name, BSSID))
6: Aggregate tuples by user MAC: ordered by timestamp
7: end if
8: end while
9: for all user MACs do
10: Parse {message subtype} $\rightarrow$ 802.11 FSM
11: Determine association and authentication per AP
12: Define presence: (start time, end time, AP name, BSSID)
13: Aggregate presences by user MAC: ordered by timestamp
14: end for

During processing of one week of log files, we were able to successfully process $\approx 95\%$ of the logged events with the FSM. Among, the 5% of events that we could not process using the FSM, we found user (dis)associations with missing (de)authentication messages. On an Intel i5 dual-core processor each day of data takes approximately a half hour to process.

4. USER MOBILITY CHARACTERISTICS
For each day of activity, we have identified on average 40,000 unique MAC addresses. Fig. 3 shows the cumulative number of unique MAC addresses and usernames seen over a five-day period. Each MAC address observed in the syslog data corresponds to a (dis)association or a (de)authentication event. The usernames are logged when a user authenticates to the wireless network. From the figure, the number of MAC addresses is seen to be nearly three times the number of usernames, showing that on average, each user accesses the network using three different devices. These devices could be either UMass devices located in computer labs, libraries, etc. or personal devices (e.g., a mobile phone, tablet and laptop).

The holding time distribution $H(t)$ at an AP is shown in Fig. 4. The distribution shows a peak for hold times $H(t) \leq 1$ minute. A large fraction of hold times, $H(t) \leq 1$ minute, are due to multiple (dis)association events to the same AP, or neighboring APs (ping-pong effects). A second peak observed at $H(t) \approx 15$ minutes corresponds to the idle-timeout of a user, if the device does not respond to an ARP (Address Resolution Protocol) request which is set by default within ARUBA. The subsequent peaks observed with $H(\Delta t) \approx 15$ minutes, occur when the user’s device responds to the ARP request just before the idle time out. The mean hold time at an AP was observed to be 12 minutes.

Each session of a user’s trajectory consists of a sequence of presences at several APs. Within each session, however, we also observe consecutive presences to the same AP. If the gap length between two consecutive presences to the same AP is less than an aggregation gap length $G_{agg}$; we aggregate the presences. The results of aggregating successive presences at the same AP are shown in Fig. 5. The figure shows the mean hop count for user trajectories as a function of the aggregation gap. As $G_{agg}$ increases, there is a decrease in the mean number of hops per session. From the
figure, we aggregate two consecutive presences at the same AP if $G_{agg} \leq 10$ minutes.

Similar to previous trace studies, we observe ping-pong phenomenon among access points [6,8,12]. We plan to work on removing ping-pong effects in future work.

5. RELATED WORK

In [6] and [8] authors present WLAN traces over two large campus environments at Dartmouth and University of California, San Diego. The UMass syslog traces provide one of the largest traces across any university campus. The (4500) access points cover the entire campus, and we observe a very large flow of students (25,000) and staff (8,000). Our observation period started in November, 2013, and the traces are continuously updated in real time.

With the changing usage of wireless access from laptops to mobile phones and tablets, the data collected also presents a different perspective on mobility from those seen in [6] and [8]. In the Dartmouth traces [6], the authors identify VoIP phones and pocket PCs in total contribute to 1.09% of the users. Since then, the number of smartphones and tablets has increased dramatically, and represent a very different usage pattern today. We plan to release a differentially private data set for use by the research community which we hope can provide valuable insights into user mobility characterization.

Acknowledgement. This research was supported in part by the US National Science Foundation awards CNS-1040781.

6. REFERENCES

[10] The Link.1 The STMS OPUS card is unsafe and unsound. NSF Workshop on Pervasive Computing at Scale, PeCS, January 2011.
Figure 3: Number of unique MAC addresses and usernames observed over a five day period.

Figure 4: Holding time distribution $H(t)$ at an AP (ping-pong effects not mitigated).

Figure 5: Mean number of hops in a user trajectory as a function of the aggregation gap length over a one-week period (ping-pong effects not mitigated).