Problems with the Dartmouth wireless SNMP data collection

Tristan Henderson, David Kotz
Department of Computer Science,
Dartmouth College
{tristan,dfk}@cs.dartmouth.edu

Dartmouth Computer Science Technical Report TR2003-480, Revision 2

December 31, 2003

Abstract

The original Dartmouth wireless network study [KE02, KE03] used SNMP to query the college's Cisco 802.11b access points. The perl scripts that performed the SNMP queries suffered from some problems, in that they queried inappropriate SNMP values, or misunderstood the meaning of other values. This data was also used in a subsequent analysis [Lee03]. This document outlines these problems and indicates which of the data collected by the original scripts may be invalid.

1 Introduction

Since its installation in 2001, David Kotz and his students have tracked the usage of the Dartmouth wireless 802.11b network by several means. They collected syslog and DHCP records, sniffed packets using tcpdump, and polled SNMP records. This paper is about the SNMP data collection methods.

To collect data using SNMP, a script polled each access point (AP) periodically, and retrieved certain SNMP values. This data includes information about the AP, such as uptime and the amount of transmitted and received traffic, and information about each client on the wireless network, such as their IP address and signal strength. Unfortunately, we recently found problems in these scripts, and this report outlines these problems and suggests limited solutions and workarounds.

2 Problem — querying the AP's forwarding table

The problems lie in the SNMP queries intended to extract information about the wireless clients. The intent of the original SNMP query script was to determine information about those clients currently associated with an Access Point (AP). All of the Dartmouth APs at the time were Cisco Aironet model 340 or 350. To collect information about the wireless clients, the script queried the `awcDot11TpFdbTable`. This table is described in the AP's relevant MIB [MIB02] as:

```
awcDot11TpFdbTable OBJECT-TYPE
                SYNTAX  SEQUENCE OF AwcDot11TpFdbEntry
                MAX-ACCESS  not-accessible
                STATUS  current
                DESCRIPTION
                        "A table that contains information about
                        entries for which the bridge has forwarding
                        and/or filtering information.  This table
                        maintains only 802.11-specific information
                        about each entry."
        ::= { awcForwardTbl 5 }
```

In other words, `awcDot11TpFdbTable` lists all the hosts in the AP's forwarding table. Since the AP typically acts as a bridge, this table will contain entries for more clients than just those that are currently associated to the AP. For instance, it may contain entries for those clients that have just left the AP.

The original scripts walk the entire `awcDot11TpFdbTable` and incorrectly assume that all the clients in this table are associated to the AP being queried. Since the table includes all of the clients in the AP's forwarding table, use of this table leads to an overestimate of the number of clients associated to an AP.

2.1 Solution

After further analysis of the MIBs, we discovered that the APs also contain counters (not recorded by the original polling scripts) that indicate the current number of clients associated with an AP. In particular, there is a counter, `awcFtClientSTASelf`, which is described in the MIB as:

```
awcFtClientSTASelf OBJECT-TYPE
                SYNTAX Integer32
                MAX-ACCESS read-only
                STATUS current
                DESCRIPTION
                        "Count of Client Stations which are Associated
                         to the system."
        ::= {  awcFtStatistics 8 }
```

`awcFtClientSTASelf` thus indicates the number of clients that are currently associated with the AP. If further information is desired about these clients, then the aforementioned `awcDot11TpFdbTable` must be queried. Care must be taken, however, to distinguish between those clients that are associated with the AP, and those that are clients for which the AP only contains forwarding information. To distinguish between these two types of clients, each table entry contains another variable, `awcDot11TpFdbAID`:

```
awcDot11TpFdbAID OBJECT-TYPE
        SYNTAX Unsigned32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
            "AID with which the Station is associated with this
         system, or 2008 if the
         Station is not currently known to be associated.  If the
         entry is multicast, awcDot11TpFdbAID is 0.  Note that
         the uplink from a Client or Repeater AP to its parent
         is always AID 1."
::= {  awcDot11TpFdbEntry 2 }
```

By checking that the value of `awcDot11TpFdbAID` is not equal to 0, 1, or 2008, it can be determined whether a client listed in the `awcDot11TpFdbTable` is associated to the AP being queried. Unfortunately, the polling scripts did not record `awcDot11TpFdbAID`.

3 Problem — AP timeouts

The original scripts queried each AP in turn, waiting for each AP to return before moving on to the next one. The scripts may take a long time to return for certain APs. This can be due to a variety of factors. The AP may be unreachable, in which case the script needs to wait for a timeout, or the `awcDot11TpFdbTable` may be very large, in which case the time taken to walk the entire `awcDot11TpFdbTable` can be very long. An AP which takes a long time to complete a query holds up the entire polling process, due to its serial nature. In the Dartmouth wireless study, each AP is polled every five minutes. It is often the case that the previous set of polls did not complete before a new set of polls was scheduled to begin, so some polls occurred later than scheduled.

3.1 Solution

To overcome some of these problems, we improved the original scripts using a variety of methods. First, rather than polling each AP in turn, the new scripts poll all the APs simultaneously, and SNMP responses from the APs trigger a callback function that processes the received SNMP data. In this way, the total time taken by the script is equal to the slowest AP, rather than the sum of all the AP queries. Second, SNMP GETBULK requests are used rather than GETNEXT. This reduces the amount of SNMP traffic, and reduces the time taken to query APs with a large `awcDot11TpFdbTable`.

4 Consequences

As the original scripts query the `awcDot11TpFdbTable` without recording the value of `awcDot11TpFdbAID`, they will tend to overestimate the number of clients associated with a given AP. Using the techniques described in 2.1 and 3.1, we developed a new set of Perl scripts to query the APs. Using two Linux hosts, all of the Dartmouth APs were queried using both the old and new scripts. Each polled every access point every five minutes for two weeks.

Figure 1 shows the total number of clients over the two-week polling period, as calculated by the new and old scripts. The old scripts (the higher line) generally overestimated the number of clients associated with each AP. The average number of clients estimated by the old scripts was 2064.17, whereas the new scripts calculate an average number of clients of 1211.61. There were also two occasions with the new scripts where the number of clients was reported to be zero, due to network problems with the host that was running the new scripts that were unrelated to the SNMP queries themselves.

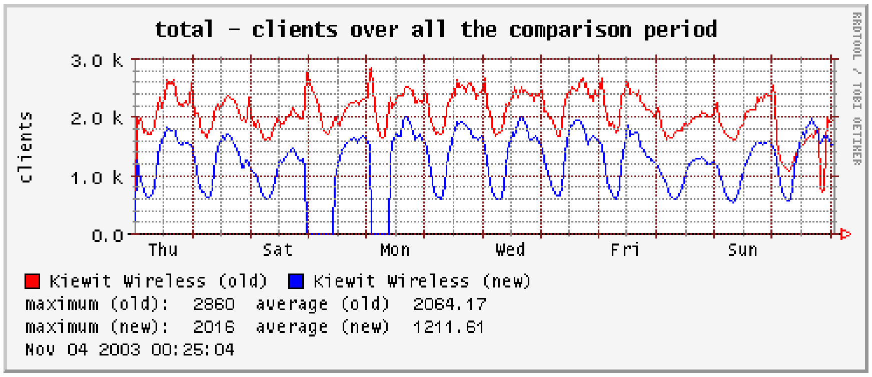

Figure 1: Total number of clients reported by old and new scripts

The old scripts did not overestimate by a constant amount or factor, and towards the end of the polling period the old scripts sometimes *underestimated* the number of clients. They underestimate because of the timeout and inefficiency problems mentioned above. Figure 2 shows the output from the old and new scripts for Berry, which is part of Dartmouth's main library and sees many mobile users. The APs in Berry thus tend to have many entries in the `awcDot11TpFdbTable`. The old scripts often timed out on retrieving values from these APs, and as a result, Figure 2 shows that the new scripts generally returned a larger number of clients than the old scripts.

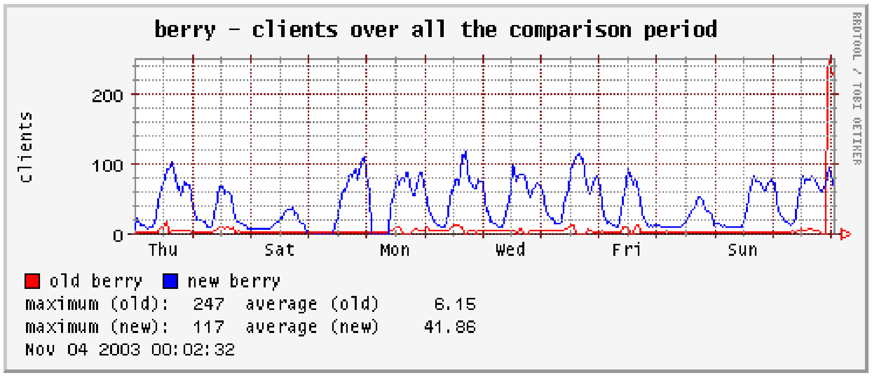

Figure 2: Total number of clients in Berry library

Overestimation by the old scripts was the norm, however. Figure 3 shows the data for a typical AP, where the old scripts estimate an average of 11.37 associated clients, whereas the new scripts only report an average of 4.51.

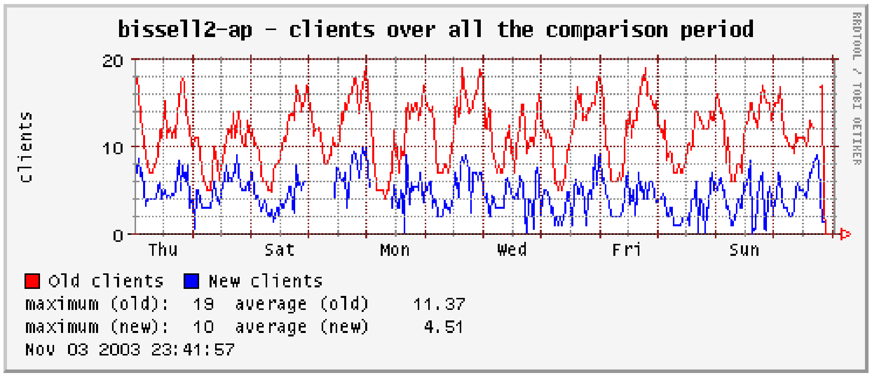

Figure 3: Total number of clients on an individual AP

As a sanity check for the new scripts, we compared the value of `awcFtClientSelf` with the number of entries in `awcDot11TpFdbTable` with an `awcDot11TpFdbAID` value not equal to 0, 1 or 2008. We found the two values to be identical 88.5% of the time. They were not always the same because the SNMP queries for these two values take place at slightly different times, and clients may associate with or leave an AP between queries.

5 Solutions for existing data

The techniques described above will only work for analysis of data collected by the new scripts, as they require new techniques and SNMP values that are not available in data collected from the old scripts. For this old data, it may still be possible to extract the aggregate number of clients on the wireless network by looking for unique users at each poll. Clients may appear in the `awcDot11TpFdbTable` on several APs, and it should be possible to discard duplicate data points. It is not possible, however, to determine to which particular AP a client is associated without knowing the value of `awcDot11TpFdbAID`. Moreover, since all APs were not polled at exactly the same time, discarding duplicate entries may be inappropriate, since a client may have changed AP in the gap between polls.

Please note that these inaccuracies only affect the client data reported by the SNMP scripts. Any AP or network interface-specific information (such as the system uptime, transmitted bytes), is still accurate.

6 Impact on [KE02, KE03]

This study made minimal use of the data from `awcDot11TpFdbTable`. We focus on the definitive journal version of the paper [KE03].

In that paper [KE03], Figure 7 derived from this data. This plot likely underestimated the traffic per card, because the number of cards was overestimated in two ways. First, the analysis counted the number of unique MAC addresses seen at each AP that day, by examining the `awcDot11TpFdbTable`. Then, counts were summed across all APs to obtain a campus-wide count, failing to account for the fact that the same MAC may have visited multiple APs, and thus counted multiple times.

In that paper [KE03], Figures 30 and 32 also derived from this data. As in the previous case, this plot likely underestimated the traffic per card, because the number of cards per category or per building was overestimated in the same two ways as above.

7 Impact on [Lee03]

All of the data in this paper is suspect, because this analysis attempts to track the arrival and departure of individual MAC addresses at individual APs, using `awcDot11TpFdbTable`. Since the table lists all MACs in the APs bridge table, it may contain MACs that never visited the AP, or may retain a MAC after it has roamed to another AP.

8 A note about [BC03]

Balazinska [BC03] used a version of our scripts, although modified to check for non-associated users (`awcDot11TpFdbAID`= 2008). It appears that her results are thus not affected.

9 Conclusion

The SNMP scripts used in the original Dartmouth wireless study to query the Cisco 802.11b APs unfortunately contain some errors. This paper outlines the problems and a possible workaround. As a result of these problems, data collected by these scripts must be used with care, and some conclusions drawn from this data may be inaccurate. We demonstrate the nature of these inaccuracies by comparing data from the old scripts to newer improved scripts.

References

[BC03] Magdalena Balazinska and Paul Castro. Characterizing Mobility and Network Usage in a Corporate Wireless Local-Area Network. In *Proceedings of the 2003 International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 303–316, San Francisco, CA, May 2003. USENIX Association.

[KE02] David Kotz and Kobby Essien. Analysis of a campus-wide wireless network. In *Proceedings of the Eighth Annual International Conference on Mobile Computing and Networking*, pages 107–118, September 2002. Revised and corrected as Dartmouth CS Technical Report TR2002-432.

[KE03] David Kotz and Kobby Essien. Analysis of a campus-wide wireless network. *Mobile Networks and Applications*, 2003. Accepted for publication.

[Lee03] Clara Lee. Persistence and prevalence in the mobility of dartmouth wireless network users. Technical Report TR2003-455, Dept. of Computer Science, Dartmouth College, May 2003.

[MIB02] Cisco Aironet wireless LAN access point MIB file, October 2002. ftp://ftp.cisco.com/pub/mibs/v2/AWCVX-MIB.my.