Characterizing the IEEE 802.11 Traffic: The Wireless Side

Jihwang Yeo, Moustafa Youssef, Ashok Agrawala
Department of Computer Science
University of Maryland
College Park, MD 20742
{jyeo,moustafa, agrawala}@cs.umd.edu

CS-TR-4570 and UMIACS-TR-2004-15
March 1, 2004

***Abstract*— Many studies on measurement and characterization of wireless LANs have been performed recently. Most of these measurements have been conducted from the wired portion of the network based on wired monitoring or SNMP statistics. In this paper we argue that traffic measurements from a wireless vantage point in the network are more appropriate than wired measurements or SNMP statistics, to expose the wireless medium characteristics and their impact on the traffic patterns. While it is easier to make consistent measurements in the wired part of a network, such measurements can not observe the significant vagaries present in the wireless medium itself. As a consequence constructing an accurate measurement system from a wireless vantage point is important but usually quite difficult due to the noisy wireless channel. In our work we have explored the various issues in implementing such a system to monitor traffic in an IEEE 802.11 based wireless network. We show the effectiveness of the wireless monitoring by quantitatively comparing it with SNMP and measurements at wired vantage points. We also show the analysis of a typical computer science department network traffic using the wireless monitoring technique. Our analysis reveals rich information about the PHY/MAC layers of the IEEE 802.11 protocol such as the typical traffic mix of different frame types, their temporal characteristics, correlation with the user activities and the error characteristics of the wireless medium. Moreover, we identify anomalies in the operation of the IEEE 802.11 MAC protocol. Our results show excessive retransmissions of some management frame types reducing the useful throughput of the wireless network. We also find that some features of the protocol, which were designed to reduce the retransmission errors, are not used. In addition, most of the clients fail to adapt the data rate according to the signal condition between them and the access point, which further reduce the useful throughput.**

I. Introduction

With the popularity of the IEEE 802.11 [1] based wireless networks, it has become increasingly important to understand the characteristics of the wireless traffic and the wireless medium itself. A number of measurement studies [2]–[7] have examined traffic characteristics in wireless networks. In these studies, the measurements have been conducted on the wired portion of the network and/or combined with *SNMP* logs [2].

The measurements at such wired vantage points can provide accurate traffic statistics as seen in that portion of the network. However they are mostly unable to expose the wireless medium characteristics (PHY/MAC in the IEEE 802.11) because they cannot observe the actual IEEE 802.11 frames on the air.

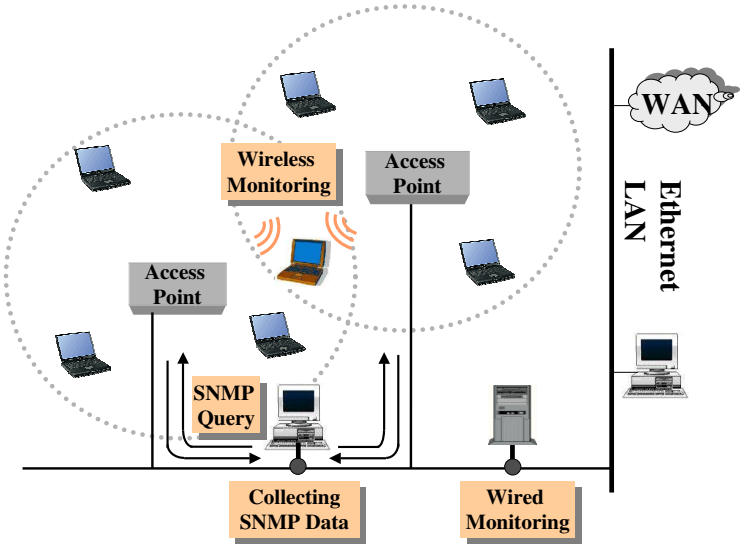

Fig. 1. Monitoring Wireless Traffic: from a *wired* vantage point, a *wireless* vantage point, and SNMP statistics.

SNMP [16], [17], [19] can be analyzed to give aggregate statistics about the IEEE 802.11 wireless LAN (WLAN) from the access point (AP) perspective. Such information includes the number of erroneous frames and reasons for most recently disassociated stations, etc. They provide accurate traffic size information at Access Points [16], [17]. However this information is either aggregate (e.g. Frame Count) or instantaneous information (e.g. last Disassociated Station) that depends on the polling interval (typically order of minutes). Therefore they hardly represent complete frame-by-frame statistics nor represent the statistics from the client point of view.

In this paper, we introduce *wireless monitoring* as a traffic characterization technique. Rather than looking at part of the picture through wired monitoring and/or SNMP statistics, we

argue that traffic measurements from a *wireless* vantage point in the network are crucial to analyzing the full picture of the 802.11 wireless network. We show that not only does wireless monitoring give the same information provided by wired monitoring and SNMP statistics, but it also provides much richer information about the wireless medium.

Fig. 1 illustrates "wired monitoring", a measurement from a *wired* vantage point, and "wireless monitoring", a measurement from a *wireless* vantage point, and SNMP statistics.

We believe that our study is the first to expose the PHY/MAC characteristics of the IEEE 802.11 traffic and to obtain more detailed error statistics than studies based on wired monitoring or SNMP statistics alone. Therefore, our work presents a basis for building models and simulation tools of the 802.11 wireless networks. Moreover, our detailed analysis of the wireless medium allowed us to *identify anomalies* in the operation of the IEEE 802.11 MAC protocol that have a large impact on the network throughput (Section V). This study would help protocol designers and manufacturers to refine the protocol and implementation to remove the anomalies identified.

We have performed a detailed passive measurement experiment over a period of two weeks, in which we have observed the wireless PHY/MAC characteristics in the A.V. Williams building on the campus of University of Maryland, which houses the Department of Computer Science. The wireless network in this building has a high traffic load. Our observations indicate that indeed a consistent inference based on a wireless measurement is possible. However, the measurement process is significantly more challenging than when performed from a wired vantage point.

A. Advantages of Wireless Traffic Monitoring

The wireless monitoring system consists of a set of devices which we call *sniffers*, to observe traffic characteristics on the wireless medium. Wireless monitoring is more useful for understanding the traffic characteristics in wireless network for the following reasons.

A wireless monitoring system can be set up and put into operation without any interference to existing infrastructure, e.g. end-hosts and network routers. In fact wireless monitoring can be performed without any interaction with the existing network, and hence is completely independent of the operational network.

More importantly, wireless monitoring exposes the characteristics on the wireless medium itself so that we can infer the PHY/MAC characteristics. Thus wireless monitoring allows us to examine physical layer header information including signal strength, noise level and data rate for individual packets. Similarly it also enables examination of the link layer headers, which include IEEE 802.11 type and control fields [1]. This is not possible with the measurements at a wired vantage point. Compared to SNMP logs, wireless monitoring allows us to have detailed information about all stations, while in SNMP logs only the aggregate or instantaneous information as seen from the AP view are available.

Physical layer information can be used to see how they correlate with error rates and throughput. This is useful for developing accurate error models for the IEEE 802.11 WLANs and in site planning to determine the minimum signal strength required to achieve a certain throughput or error rate.

By analyzing the link layer data, we can characterize traffic according to different frame types, namely: data, control, and management frames.

The collected data, combined with timestamps, can be used as accurate traces of the IEEE 802.11 link-level operations. Such traces are useful when we want to emulate the protocol or diagnose problems of wireless networks.

Several throughput models [8], [9] on the IEEE 802.11 have been introduced, which propose collision rate, transmission rate and throughput as important performance metrics. Wireless monitoring enables us to make exact measurement of such IEEE 802.11 MAC-level performance metrics.

B. Challenges of Wireless Monitoring

The advantages we mentioned above, however, would not be exploited unless the sniffers can capture nearly all the frames on the air. Unfortunately it is very difficult to guarantee that any sniffer can capture all such wireless frames. We have observed that typically most of these losses are due to signal strength variability, card variability or a combination of both. Losses in the sniffers pose the most challenging problem in wireless monitoring.

If sniffer losses are inevitable, then the following questions can be raised. How can we reduce such sniffer losses? How can we justify that even with such losses the measurement provides meaningful results, consistent with the end-to-end real world experiences? In order to answer these questions, we conducted a controlled experiment using an end-to end measurement tool as a baseline for accuracy. In Section III we present the results for this experiment that helped us identify the pitfalls that a wireless measurement system needs to be aware of. We also present the techniques that can be used to avoid them. These techniques can be used in future wireless monitoring-based experiments.

C. Organization

The rest of the paper is organized as follows. In Section II we discuss previous works in the area of WLAN traffic characterization. Section III describes the controlled experiment, the pitfalls of wireless monitoring, techniques to overcome them, and how wireless monitoring compares to wired monitoring and SNMP statistics. We describe the results of our two-week long experiment in Section IV. In Section V we discuss the anomalies we discovered in the 802.11 implementations and the traffic characterization. Finally, we conclude the paper in Section VI and highlight our ongoing work.

II. Related Work

Several measurement and analysis studies [2]–[4], [7], [11] have examined traffic or error characteristics in the IEEE

802.11 WLAN. Most of the measurements have been performed on university WLAN [3], [4], [7], [11], while the work in [2] examined WLAN traffic in a conference environment.

The study of Tang and Baker [4] in the Computer Science Department building of Stanford University was one of the early studies. They examined wired monitoring traces, and SNMP logs to analyze a twelve-week trace of a local-area wireless network.

In a public-area wireless network, the traces collected in well-attended ACM conference were successfully analyzed by Balachandran et al. [2]. They used SNMP logs and wired monitoring to characterize not only the patterns of WLAN usage, but also the workloads of user arrivals and session durations with parameterized models. They also analyzed channel characteristics using SNMP logs and presented ***aggregate*** error percentages from the AP point of view.

A significantly larger scale experiment covering a much longer duration and coverage area has been presented in the Dartmouth campus by Kotz and Essien [3]. Their analysis was based on using system logs in APs, SNMP logs and wired monitoring traces to characterize the typical usage and traffic patterns in a university WLAN.

In a similar recent study, Schwab and Bunt [7] used wired monitoring and the Cisco proprietary LEAP authentication logs to characterize one-week usage and traffic patterns in a campus-wide WLAN environment.

Similar to the previous studies, our measurements are performed on typical university WLAN environment in a department network. We are interested in showing the traffic characteristics for a typical access point in this environment. Our *uniqueness* comes from analyzing the wireless media using the ***wireless monitoring*** technique which gives a full view of the network spanning all the layers of the protocol stack. In all the above studies the measurements have limited analysis of the PHY/MAC layer based on the aggregate data of the SNMP traces.

In a more general wireless environment, the authors in [5], [6] performed wireless monitoring to measure packet loss and Bit Error Rate. Their experiments were fully controlled between two wireless stations and performed on non-802.11 networks. Our work is different in being in the context of 802.11 WLANs and in performing the experiment in an actual environment with different goals.

III. Controlled Experiment

In this section, we present our controlled experiment. The purpose of this experiment is to analyze the wireless monitoring technique in terms of its effectiveness in capturing wireless traffic and presenting precise statistics for wireless medium. Moreover, we compare its performance to that of wired sniffing and SNMP statistics. Since ***only*** the wireless monitoring provides detailed information about the PHY/MAC layer, we base our comparison in this section on the percentage of frames that can be captured by different techniques compared to the frames generated by a reference application.

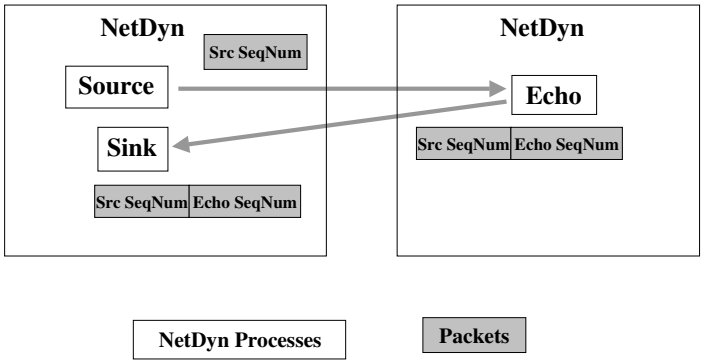

Fig. 2. NetDyn processes and their sequence numbers.

For more detailed description of our controlled experiments readers are recommended to refer to Appendix II.

A. Methodology

1) Network Infrastructure: We perform our experiments in the A.V. Williams building, at University of Maryland (where the Department of Computer Science is located). The building has 58 access points installed, which belong to three different wireless networks. Each wireless network is identified with its *ESSID*. The ESSIDs of the three networks are *umd*, *cswireless* and *nist* respectively. *umd* network consists of 29 Cisco Aironet A-340 APs, the most widely used wireless network in the university. *cswireless* (12 Lucent APs) and *nist* (17 Prism2-based APs) are built by individual research research groups in the department[1].

We performed our controlled experiment on a separate network that we set up specifically for this purpose with its own ESSID. Our clients were configured to associate with this AP.

2) NetDyn: To estimate the exact measurement loss, we need to use reliable application generated sequence numbers. We conducted a two-way UDP packet exchange experiments using an end-to-end traffic measurement tool, called *NetDyn* [13].

As shown in Fig. 2, NetDyn consists of three different processes, *Source*, *Echo* and *Sink*. *Source* puts a sequence number in the payload, sends the packet to *Echo*, which also adds a sequence number before forwarding it to *Sink*. In our setup, *Source* and *Sink* processes run on a wireless station, while the *Echo* process runs on a server wired to the LAN. Using the sequence numbers generated by the *Source* and *Echo* processes, we can determine which packets were lost in the path from the *Source* machine to the *Echo* machine and vice versa.

In the experiment, *Source* sends 20000 packets with the full UDP payloads (1472 bytes) to *Echo*, with 10 ms inter-packet duration (hence, at 100 packets/second). We made sure that no fragmentation occur on either side of the AP.

[1]All networks mentioned in the paper are based on the 802.11b protocol.

Therefore, for each NetDyn frame on the wireless side, there is a corresponding frame on the wired side and vice versa. We use the NetDyn statistics as the baseline for comparison with other monitoring techniques.

3) Monitoring Hardware/Software: We set up three sniffer machines to capture the wireless frames on the air. All sniffing devices use the Linux operating system with kernel version 2.4.19. We used *Ethereal* (version 0.9.6) and *libpcap* library (version 0.7) with the *orinoco_cs* driver (version 0.11b), patched to enable monitoring mode, as our sniffing software. We made use of the 'monitor mode' of the card to capture 802.11 frame information including the IEEE 802.11 header as well as physical layer header, called the *Prism2* monitor header, and higher layer protocols' information.

A wired sniffer was installed on the same LAN as the AP and the NetDyn *Echo* machine through a *Century Tap*, a full-duplex 10/100 Ethernet splitter [20]. The sniffer machine was running *Ethereal*. The same machine was running the SNMP client that was configured to poll the AP for SNMP statistics every one minute.

4) Captured Wireless Data : The wireless sniffer captures the first 256 bytes of each receiving 802.11 frame, records the complete view of the frame, i.e. PHY/MAC/LLC/IP/Above-IP information.

Prism2 monitor header is not a part of IEEE 802.11 frame header, but is generated by the firmware of the receiving card. The header includes useful PHY information, such as MAC Time, RSSI(Received Signal Strength Indication), SQ (Signal Quality), Signal strength, Noise and Signal Noise Ratio (SNR) and Data rate (in Mbps). All signal and noise information are in manufacture-specific units. However, they can be used for relative comparison.

We also capture the IEEE 802.11 MAC frame structure which incorporates the following fields: protocol version, frame type (management, data and control), Duration for Network Allocation Vector (NAV) calculation, BSS Id, Source and Destination MAC addresses, fragment, sequence number among others [1]. According to the 802.11 MAC frame type of the captured frame, we extract different information. For example, for Beacon frames, captured information include 64-bit Beacon timestamp which we use for time synchronization among multiple sniffers (Section III-C). For Association/Disassociate and Authentication/Deauthentication frames, the information includes the reason code for such actions.

We also capture higher layer protocol information, mainly for NetDyn frames.

For SNMP, we can capture the same statistics as in [2]. For wired sniffer data, we capture enough information to give us the NetDyn sequence numbers.

5) Experiment Setup: We tried different scenarios for the traffic between the wireless clients and the wired server. In the rest of this section we show the results of one experiment whose configuration is shown in Fig.3. Other configurations gave comparable results. We have two wireless clients at two different locations corresponding to two different signal conditions. The "Good" client lies in an area of good AP

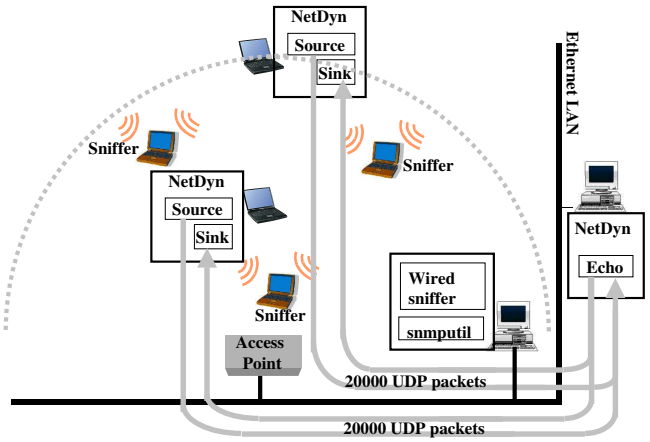

Fig. 3. Controlled Experiment using NetDyn: *Source* in a wireless station sends 20000 UDP packets to wired *Echo* machine which sends them back to *Sink* in the same wireless station.

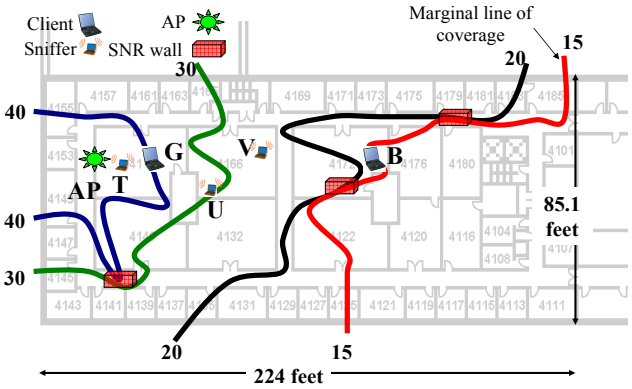

Fig. 4. SNR Contour Map for controlled experiment: SNR Contour lines for 40,30,20 and 15 dB are obtained from SNR measurements. Based on the contour map, we place the wireless clients at locations G and B and place the sniffers at locations T, U, and V.

coverage, in terms of SNR, while the "Bad" client lies in an area of bad AP coverage. We also have three wireless sniffers (T, U and V) capturing the wireless traffics between *Source*, *Sink* and the AP. Sniffer T is placed adjacent to the AP while the other two sniffers are placed as shown in Fig.34[2]. Note that the purpose of placing the sniffers in the controlled experiment was not to maximize the capture performance, but rather to study the different factors affecting the wireless monitoring performance.

B. Single Sniffer Statistics

We define a "From-AP" frame, as a frame transmitted by the AP to a wireless station. Similarly, we refer to a frame from the wireless station to the AP as a "To-AP" frame. Table I shows the number of received packets for the NetDyn application and the percentage of MAC frames captured by the three wireless sniffers. We define the *measurement loss* to be the percentage of the frames *unobserved* by the sniffer.

[2]We discuss sniffers placement in Section III-D.

The entries for the wireless sniffers were obtained by counting all frames with ***unique sequence numbers***. We can make the following observation from the table:

- Different sniffers have different viewpoints of the wireless medium.
- The percentage of measurement loss for From-AP traffic is much less than the percentage of measurement loss for To-AP traffic. On the average, one sniffer can see 99.4% for From-AP traffic and 80.1% for To-AP traffic. The reason for that is that the AP has better hardware compared to clients, therefore the signal seen at a sniffer from an AP is stronger than the signal seen from a client. Moreover, we can always place a sniffer adjacent to an AP , whose position is fixed, while we cannot do that for wireless clients as their position is not known in advance.
- Each sniffer has a significant percentage of unobserved frames compared to NetDyn data. Even sniffer T, which was placed adjacent to the AP, encountered a severe measurement loss to observe only 73% of the total traffic. These measurement losses may be due to signal strength variability, card variability or a combination of both.
- The *absolute* physical location of the client or the sniffer does not affect the ability of a particular sniffer to capture data from a particular wireless client. Rather, the *relative* position between the wireless client and the sniffer is the factor that affects the ability of a sniffer to capture the data from that client. For example, for the traffic originating from Bad client, sniffers U and V capture more traffic than sniffer T. The reason for that is as the distance from the sniffer to the wireless client increases, the signal strength decays and the SNR decreases leading to worse signal conditions and decreased sniffing performance. Sniffers U and V are closer to Bad client than sniffer T.
- In Bad client case, the sniffers captured some frames that was not received by the NetDyn application (capture percentage $>$ 100%). This is because all sniffers are closer to the AP than Bad client which means that a frame sent by the AP will have a better SNR at the sniffer compared to Bad client. Therefore, the sniffers can capture frames that Bad client cannot capture.

From these observations we can see that two factors are important to achieve a good capture percentage, i.e. a low measurement loss, from wireless monitoring:

1) Merging the data collected from different sniffers to obtain a better view of traffic.
2) Carefully selecting the sniffers location to obtain an acceptable capturing performance.

We address the two factors in the next sections. Moreover we introduce two techniques for improving the performance of wireless monitoring, namely *merging multiple sniffers* and *sniffer placement*. We briefly describe those techniques in the following sections. For more detailed description of those techniques, readers are recommended to refer to Appendix I.

C. Merging Multiple Sniffer Data

The main problem that needs to be tackled in order to merge the data from different sniffers is how to synchronize the traces when each of them is time-stamped according to the local clock of the sniffer. In this section, we describe our method for time synchronization, merging procedures and the effect of merging respectively.

1) Time Synchronization between Multiple Traces : To correctly merge multiple sniffers' data without reordering we require the time synchronization error (the difference between two timestamps of different sniffers for the same frame) to be less than the minimum gap between two valid IEEE 802.11 frames. In the IEEE 802.11b protocol, the minimum gap, G_{min}, can be calculated as the 192 microsecond preamble delay plus 10 microsecond SIFS (Short Inter-Frame Space), a total of 202 microsecond.

Our approach is to use the IEEE 802.11 Beacon frames, which are generated by the AP, to be the common frames to all the sniffers. Beacon frames contain their own 64-bit absolute timestamps as measured by the AP, therefore we can uniquely identify such common beacon frames in different sniffer traces. With such n common beacon frames, we then take one of the sniffers as a reference point and use linear regression to fit the other sniffers' timestamps [3] to the reference sniffer.

Fig. 33 shows the fitting error (difference between the fitted timestamp and the reference timestamp) for the common Beacon frames over a 12.5 minutes interval. During this period, there were 5658 Beacon frames that were common to all the sniffers out of the total of the 7500 total Beacons frame that are sent at the 100 ms rate. Sniffer T was taken as the reference sniffer in this experiment. We can see that the maximum error is below 40 microseconds, well below the 202 microseconds limit.

2) Merging Procedures : Using the obtained linear equation, we can convert the timestamp of each frame captured by each sniffer, to the reference time. To identify the duplicate frames that multiple sniffers commonly observed, we compare the header information of the frames, which are from different sniffer traces and whose converted timestamps differ by less than the minimum gap G_{min}. After removing the duplicates, we can generate a single correctly-ordered trace from multiple sniffer traces.

3) The Effect of Merging : Table I shows the effect of using the merged sniffers' traces. We can see from the table that increasing the number of merged sniffers' traces from one to two to three increases the percentage of captured frames significantly from 73.25% to 84.47% to 99.34% respectively for the To-AP traffic. Notice also that the effect of merging is more significant in the case of To-AP traffic while a single sniffer near the AP (sniffer T) can almost capture all the From-AP traffic (improvement from T only to T+U+V is 0.7%).

[3] We use the MAC time of the received frame, which is available in Prism2 header in the captured frame, as the local timestamp at each machine. We do not use the timestamp generated by the sniffer's operating system to minimize the variance of the local time measurement.

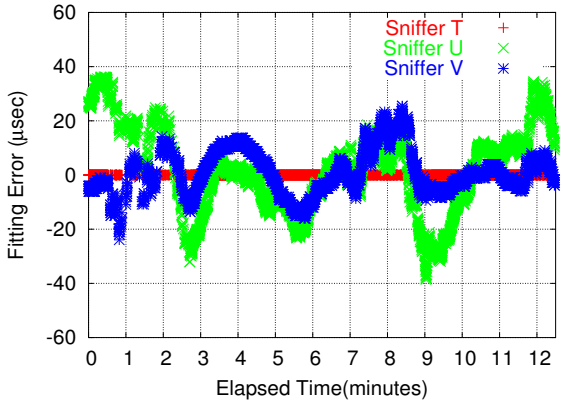

Fig. 5. Fitting error with 5658 common Beacon frames (timestamp of sniffer T is the reference time).

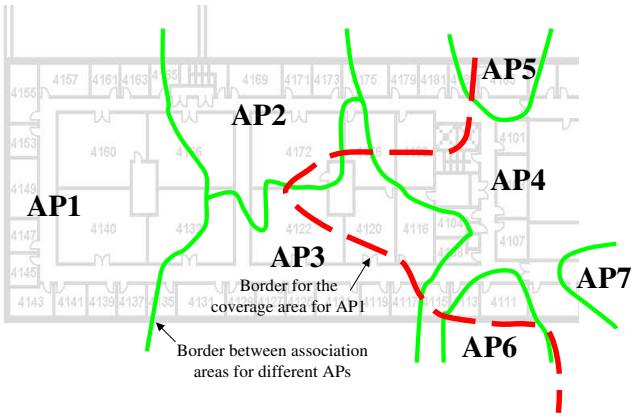

Fig. 6. The Association Area for different access points. The figure also shows the coverage area for the first access point.

Using the merged three-sniffers ' trace, wireless monitoring can capture more than 99.34% of the wireless traffic.

D. Sniffers Placement

As noted in Section III-B, carefully selecting the sniffers location is important to obtain an acceptable capturing performance. In this section, we describe our sniffer placement strategy in the coverage area of an AP . We make use of the observations presented in Section III-B.

Since in the infrastructure mode of the 802.11 protocol all traffic goes through the AP, one may think that placing all sniffers near the AP should maximize the capture performance. However, our experiments showed that the capture performance of To-AP traffic is worse than that of the From-AP traffic, even for the sniffer T which was adjacent to the AP. This is due to the weak signal that reaches the sniffer from the clients compared to the strong signal that reaches the same sniffer from the AP. The AP can capture the weak signal due to its better hardware and specialized processing (compared to the sniffer configuration).

Therefore, for placing the wireless sniffers, we should only place *one sniffer* adjacent to the AP to be responsible for capturing the From-AP traffic and the traffic of clients near the AP. Other sniffers should be placed as close as possible to the wireless clients.

If we assume that clients are going to be uniformly distributed over the coverage area, this translates to placing the sniffers so that they cover as much as possible from the AP coverage area. Therefore, if we have n sniffers to place, we can split the AP coverage area into n equal areas and place the sniffers in the center of mass of these areas.

We can refine this strategy by noting that, in an environment where multiple APs are installed, the coverage area of an AP may be reduced to the *Association Area* of the AP. The *Association Area* of an AP is the area at which a client will favor this AP for association compared with other APs in the area. Note that the Association area is a sub-area of the coverage area and that most of the traffic an AP receives comes from the associated clients (i.e. from the Association Area). Therefore, we should use the association area of an AP rather than its coverage area. Fig. 6 shows the Association Areas for different access points in the area of interest. The figure also shows the difference between the coverage area and the association area for AP_1

Another factor that needs to be taken into account is the signal condition at the sniffer location. We define an *SNR wall* as an area where the SNR contour lines are close to each other (Fig. 34). Our experiments shows that placing a sniffer near an SNR wall leads to worse capture performance compared to placing the sniffer at other places. Therefore, SNR walls should be avoided.

TABLE III
TOTAL NUMBER OF RETRANSMISSIONS

	To-AP	*From-AP*		
	Wireless	Wireless	MIB-I	MIB-II
Good	576	386	N/A	N/A
Bad	5121	4181	N/A	N/A
Total	**5697**	**4567**	**3007**	**4874**

E. Comparison Between Different Characterization Techniques

Tables XI and III show a comparison between the three traffic characterization techniques taking NetDyn results as the baseline. Note that for SNMP statistics, we based our analysis on MIB-I counters as in [2] and on the MIB-II counters.

From the table, we can make the following observations:

- Wireless monitoring has comparable performance to the other techniques for the *common* information that can be captured by other techniques.
- SNMP statistics cannot reveal per client information.
- Wired monitoring can give accurate To-AP information about the wireless medium for the successfully transmitted frames as the probability of the loss on the wired medium is order of magnitudes less than the probability of loss on the wireless medium. However, if the frames

TABLE I

INCREASING CAPTURED FRAMES BY MERGING MULTIPLE SNIFFERS: MERGING TWO OR THREE SNIFFERS AMONG T, U AND V SIGNIFICANTLY INCREASES THE NUMBER OF OBSERVED FRAMES.

	To-AP Wireless Traffic							
	NetDyn	*T*	*U*	*V*	*T+U*	*T+V*	*U+V*	*T+U+V*
Good	19905	76.76%	69.00%	68.34%	76.83%	70.00%	76.84%	98.61%
Bad	18490	69.48%	99.58%	99.73%	99.05%	100.05%	99.97%	100.13%
Total	**38395**	**73.25%**	**83.73%**	**83.46%**	**87.54%**	**84.47%**	**87.98%**	***99.34%***
	From-AP Wireless Traffic							
Good	19247	98.41%	97.31%	95.24%	99.37%	98.06%	99.32%	99.38%
Bad	17858	102.04%	101.85%	102.2%	102.56%	102.43%	102.52%	102.56%
Total	**37105**	**100.15%**	**99.5%**	**98.59%**	**100.91%**	**100.16%**	**100.86%**	***100.91%***

TABLE II

QUANTITATIVE COMPARISON BETWEEN END-TO-END MEASUREMENT USING NETDYN, SNMP, WIRED MONITORING AND WIRELESS MONITORING

	To-AP Wireless Traffic				
	NetDyn	Wireless Monitoring (%)	Wired Monitoring (%)	MIB-I (%)	MIB-II (%)
Good	19905	98.61	100.00	N/A	N/A
Bad	18490	100.13	100.00	N/A	N/A
Total	**38395**	**99.34**	**100.00**	**100.23**	**100.23**
	From-AP Wireless Traffic				
Good	19905	99.38	103.41	N/A	N/A
Bad	18490	102.56	103.53	N/A	N/A
Total	**38395**	**100.91**	**103.47**	**102.02**	**99.96**

are fragmented on the wireless medium, we cannot get correct statistics on the wireless frames from the wired side.

- For the From-AP traffic, although wired monitoring can give per client information for the wired segment, its statistics *overestimates* the actual traffic more than the wireless monitoring technique. The reason for that is the noisy characteristic of the wireless channel that leads to the loss of many packets on the wireless side that wired monitoring cannot capture.
- Only wireless monitoring can capture the retransmission information per client.
- Wireless monitoring is more accurate than the MIB-I based method that was used in [2] in characterizing the number of retransmissions.
- It is interesting to notice that even the SNMP statistics may be slightly off from the true image of the wireless client. For example, in Table XI the MIB-II total number of successfully transmitted packets is less than the number of packets received by the NetDyn Sink. This can be explained by noting that there may be packets that was successfully received by the NetDyn Sink after three retransmissions and the corresponding MAC-level ACK sent back. However, the ACK was not received by the AP and the AP did not count it as a successful transmission.

What we are trying to argue here is that wireless monitoring has comparable performance to the other techniques but has the advantage of exposing the full wireless medium frames. The next section shows the characterization for the traffic in a computer science department environment for a period of two weeks based on the wireless monitoring.

For more detailed results and discussions on the comparison, readers are recommended to refer to Appendix III.

IV. WIRELESS TRAFFIC CHARACTERISTICS: ANALYSIS OF TWO-WEEK TRACES

We apply the wireless monitoring technique to measure and characterize actual wireless LAN traffics of a typical AP in a computer science department network. We have performed passive measurements over a period of two weeks from Monday, February 9 to Sunday, February 22 to observe the wireless PHY/MAC characteristics in the fourth floor of the A.V. Williams building of Department on the campus of University of Maryland.

A. Methodology

1) Target Traffic: In the fourth floor of A.V. Williams building, we have three channel-6 APs, three channel-1 APs and one channel-1 AP. Channel 6 is the most widely used in the fourth floor, therefore we choose channel 6 as our target channel. We choose one of the channel-6 APs in the fourth floor as *our target AP*. We can monitor the traffics of both the target AP and the channel 6 at the same time, because once we set the sniffers' channel to 6, then the sniffers capture all the observed traffic on the channel (and adjacent overlapping

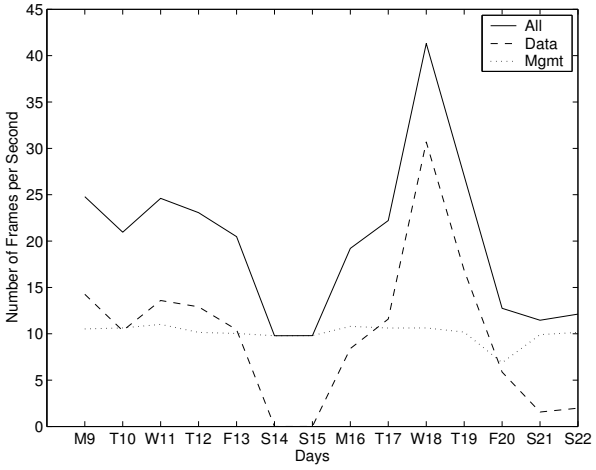

Fig. 7. [MAC Traffic] Number of MAC frames per seconds, averaged daily, over two weeks: *All* traffic of the target AP is the sum of *MAC Data* traffic and *Mgmt* (Management) traffic.

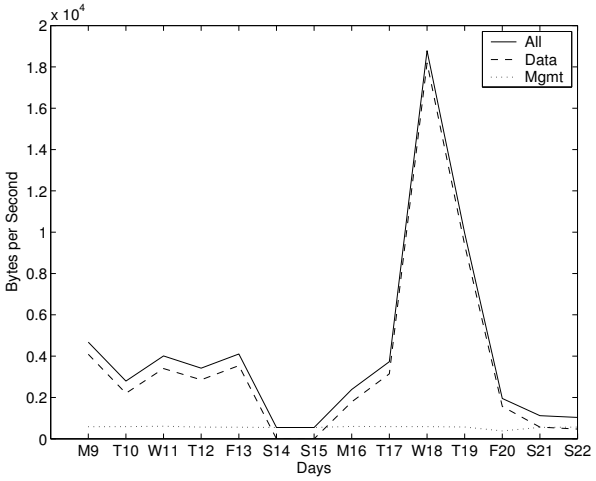

Fig. 8. [MAC Traffic] Traffic volume per seconds. Daily averaged values are shown over two weeks: *All* traffic of the target AP is the sum of *MAC Data* traffic and *Mgmt* (Management) traffic.

channel). Due to space constraints, we only show the traffic analysis for one AP in this paper.

2) Setup and Placement: The setups for H/W and S/W in three sniffers are exactly the same as in controlled experiment in Section III. We also followed our strategy for sniffer placement as discussed in the Section III-D. Sniffer *T* is the sniffer placed adjacent to the target AP.

3) Trace Collection: We started the wireless monitoring at midnight, Monday February 9, ended at midnight, Sunday February 22, for a total of 14 days.

The size of the three trace files, which are generated by *Ethereal*, is in total 12 GB in compressed format. We had a gap of 8 hours, noting that unfortunately from 4:00 pm to 11:59 pm on February 20, we were unable to capture the traffic due to disk fullness of three sniffer machines. On that day, traffic volume of channel 6 became tremendously higher than expected.

B. Results

We focus our presentation on the characterization of PHY/MAC layer traffics, which is unique to wireless monitoring. Specifically we will present our results under five categories: *MAC Traffic*, *Transmission Errors*, *MAC Frame Types*, *MAC Frame Size* and *PHY Information*. We also summarize the anomalies we discovered at the end of the section.

*1) **MAC Traffic**:* Fig. 7 and Fig. 8 show the daily traffic over the two weeks. We obtain MAC type and size information from each frame's MAC header in the traces. We separately present traffic for IEEE 802.11 Data frames and that for the IEEE 802.11 Management frames (e.g. Beacon and Probe Request frames).[4]

We notice that there was almost no user activity on the weekend of Feb. 14 and Feb. 15. This weekend represents the weekend for Valentine's Day. February 16 was the holiday for President's Day. However, the university is open in that day.

Typically traffic for the IEEE 802.11 Management frames is constant over the two weeks period. On Friday, February 20, disk space had been full for 8 hours therefore we have smaller number of frames than normal days (about one third of the normal management traffic volume).

We observe a sudden spike of traffic on Wednesday, February 18, which is three times larger than normal days. Carefully examined, we found that 40% of MAC Data traffic consists of *IMAP* (Internet Message Access Protocol) frames. IMAP protocol is used when client STA accesses electronic mail or bulletin board messages that are kept on a (possibly shared) mail server [21].

In other days, for example on another Wednesday, February 11, the traffic contains IMAP frames less than 1% (Figure 9).

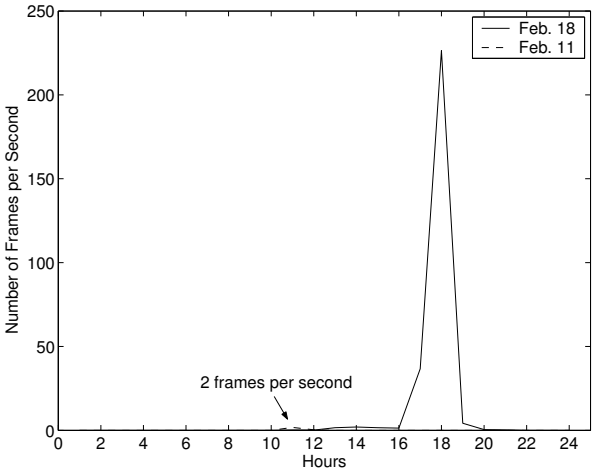

Fig. 9. [MAC Traffic] Number of **IMAP** frames per second. Plot shows hourly averaged values on February 18 and on February 11 respectively.

[4]Since the IEEE 802.11 control frames (e.g. Acknowledgement) have no *BSSID*, i.e. MAC address of AP, we do not present the results for the control frames in this section.

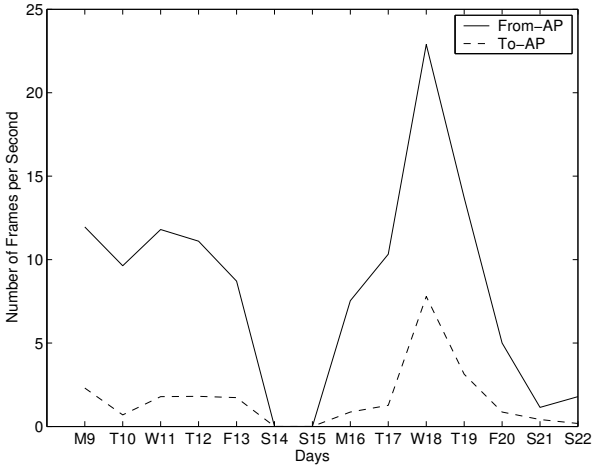

Fig. 10. [MAC Traffic] Number of **Data** frames per second. Daily averaged values are presented over two weeks.

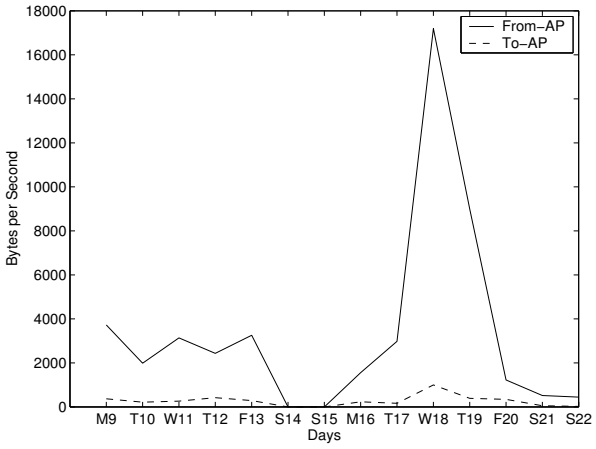

Fig. 11. [MAC Traffic] Traffic volume per second for **Data** frames. Daily averaged values are presented over two weeks.

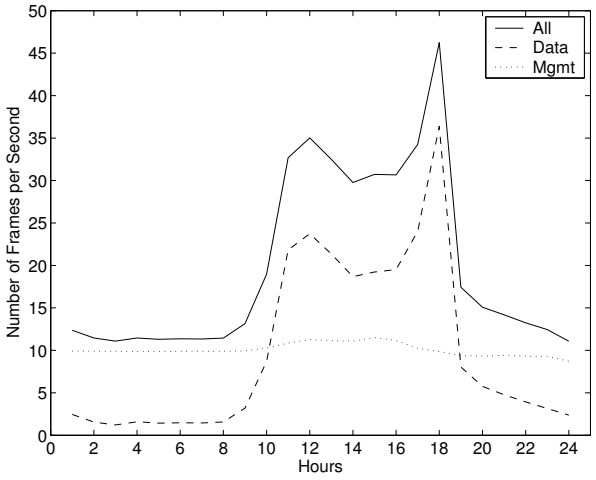

Fig. 12. [MAC Traffic] Number of frames per second. Plot shows hourly averaged values over two weeks.

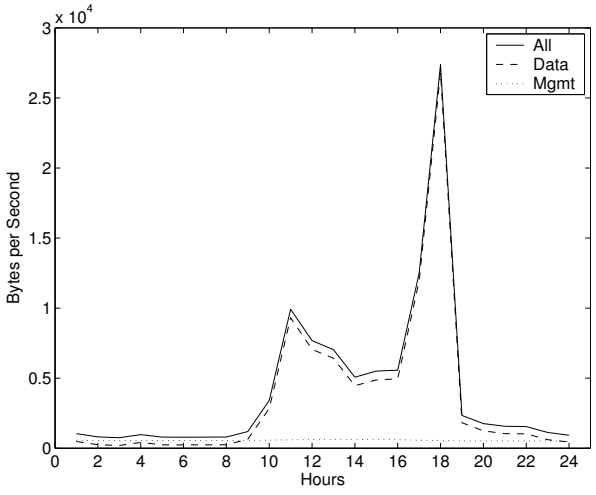

Fig. 13. [MAC Traffic] Traffic volume per second. Plot shows hourly averaged values over two weeks.

This abnormal spike of email traffic is due to email worm *W32.Netsky.B@mm* that was spreading on the web on Feb. 18 [24].

Another observation from figures 7 and 8 respectively is that the maximum throughput (in bytes per second) does not exceed 1.5 Mbps (Megabits per second). This low throughput can be explained by noting that there are 2 other APs assigned to channel 6 in our environment. These APs along with the clients associated with them, contended for the same channel with our target AP, hence reduced the throughput. We believe that this multiple-APs assignment to the same channel is a typical real world scenario.

We present the number of frames per second and the traffic volume, in bytes, for the **Data** traffic only in Fig. 10 and Fig. 11 respectively. We can observe that From-AP traffic and To-AP traffic show the same shape, which means that most of the traffic consists is two-way pairs, e.g. Request/Response interactions. For the number of frames per second, From-AP traffic has on average five times many frames than To-AP traffic. In addition, the bytes per second of From-AP traffic are roughly 12 times larger than To-AP traffic, on average. This behavior is expected as most requests are short (e.g. HTTP Get request) while the responses are larger in general.

Fig. 12 and Fig. 13 show hourly traffic variability, averaged hourly over two weeks. We can observe that the traffic has two peaks at 11 am and 6 pm. However, the peak at 6 pm is due to the abnormal high traffic volume on Feb. 18. During low user traffic periods, from about 8pm and 8am, Management frame traffic is dominant.

Fig. 14 and Fig. 15 show hourly traffics of the IEEE 802.11 **Data** frames, hourly-averaged over two weeks. We can also observe the same shapes of From-AP and To-AP traffics, which is due to the two-way interaction between clients and remote server.

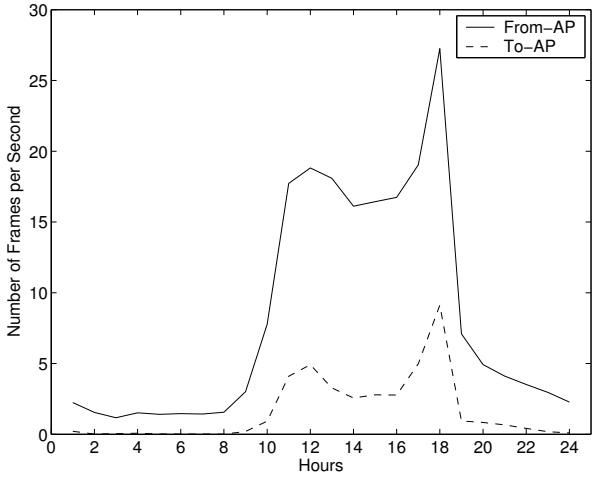

Fig. 14. [MAC Traffic] number of **Data** frames per second. Plot shows hourly averaged values over two weeks

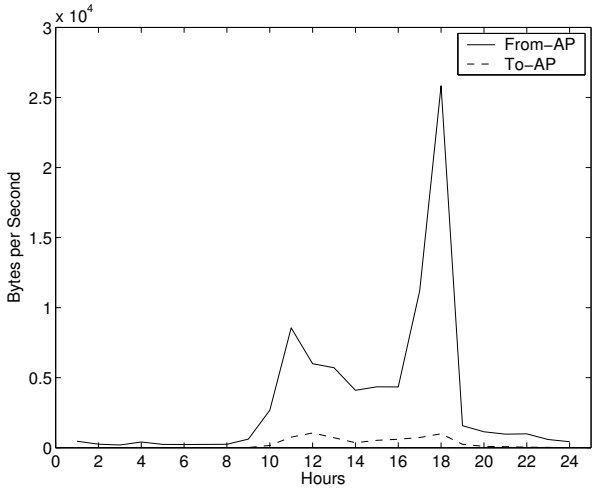

Fig. 15. [MAC Traffic] bytes of **Data** frames per second. Plot shows hourly averaged values over two weeks

*2) **Transmission Errors**:* Transmission Errors can be obtained by the number of retransmitted frames divided by the number of all frames. We can identify the retransmitted frames by examining MAC retry field in the IEEE 802.11 MAC header.

In Fig. 16 we observe that transmission errors have a high **daily** variability over two weeks. We can also observe that typically more errors occur in To-AP traffic compared to From-AP traffic. The reason is that the access point has better wireless hardware compared to clients' cards.

In Fig. 17 and Fig. 18, we show the transmission error, averaged over 10 minutes interval, for Feb. 18 for From-AP and To-AP traffic respectively. We observe that To-AP traffic shows higher variability of transmission errors as well as higher values than From-AP traffic.

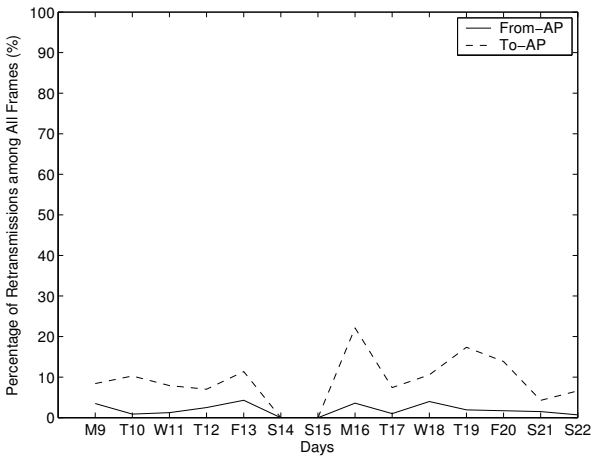

Fig. 16. [Transmission Errors] Transmission errors of **Data** frames, defined to be the number of retransmissions divided by the number of Data frames. Plot shows daily averaged values over two weeks.

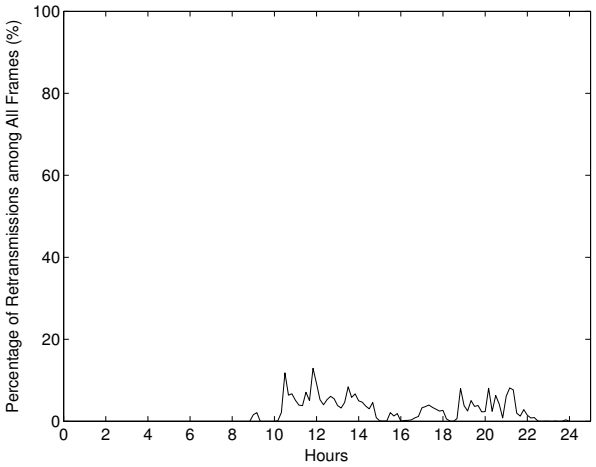

Fig. 17. [Transmission Errors] Transmission errors for From-AP **Data** frames. Plot shows the values averaged during **10 minutes** on Feb. 18.

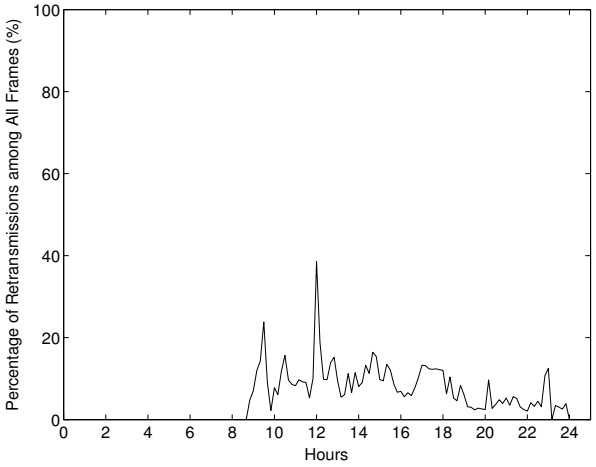

Fig. 18. [Transmission Errors] transmission errors of To-AP Data frames. Plot shows the values averaged during **10 minutes** on Feb. 18.

TABLE IV

ABBREVIATION FOR THE IEEE 802.11 TYPES

Abb.	*802.11 Types*
ACK	Acknowledgement
ProbeReq	Probe Request
ProbeRes	Probe Response
PowerSave	Power Save Poll
AsscReq	Association Request
AsscRes	Association Response
ReAsscReq	Reassociation Request
ReAsscRes	Reassociation Response
Auth	Authentication
Deauth	Deauthentication
RTS	Request-to-Send
CTS	Clear-to-Send

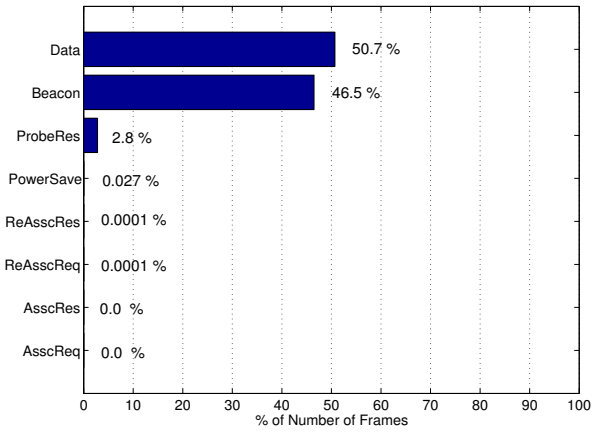

Fig. 19. [MAC Frame Types] Percentage of frames per MAC Type.

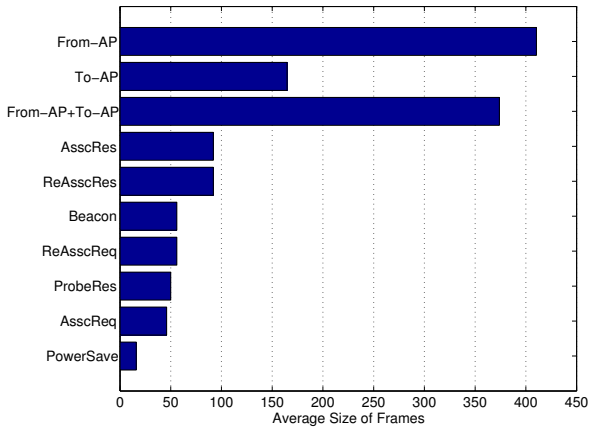

Fig. 20. [MAC Frame Types] Average frame size per MAC Type.

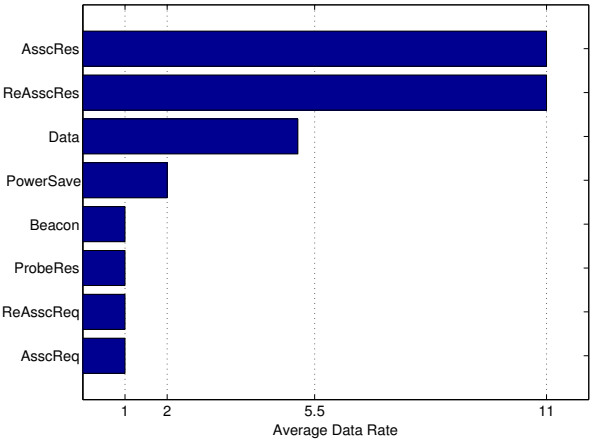

Fig. 21. [MAC Frame Types] Average data rate per MAC Type.

*3) **MAC Frame Types**:* In this section we show the results of per frame-type statistics over two weeks. For each type of frames we observed, we show the number of frames, average bytes per frame, average data rates per frame and average retransmissions per frame, respectively. We obtain this information from the 256 bytes MAC header of the IEEE 802.11 frames and the *Prism2* header which is generated per frame by the sniffer device driver (PHY information).

We use the abbreviation in Table IV to denote the long type names.

In Fig. 19 we show the number of frames per each types. Data frames are 50.7 % of the total frames, dominant in terms of the number of frames while Beacon frames are 46.5 %, dominant among management frames. We also observe that there are roughly one million Probe Response frames observed during the 14 days.

Fig. 20 shows the average frame size for each MAC type. The average size of Data frames is 374 bytes and is different for From-AP traffic and To-AP traffic (410 and 165 bytes respectively). This indicates that large frames are dominant in From-AP traffic while small size traffics are dominant in To-AP traffic as noted before.

According to the standard, some management frames may have variable size. For example, Beacon frames may have a different size according to the size of the Traffic Indication Map [1].

In Fig. 21, we have two observations:

1) AsscRes and ReasscRes are usually transmitted using the highest data rate, i.e. 11 Mbps, while the corresponding Request frames use the lowest data rate, i.e. 1 Mbps. This is not expected as the AP should respond with a data rate close to the data rate of the request to enhance the SNR at the requesting client. We discuss this observation in the next section.
2) The average data rate for Data frames is 5.1 Mbps. This strongly indicates that multiple data rates are used. We examine the distribution of the number of clients per data rate in the following section.

Fig. 22 shows the average of number retransmissions per frame. We calculate the numbers by dividing the number of retransmitted frames (whose MAC Retry bit is set to 1) by the number of non-retransmitted frames (whose MAC Retry bit is set to 0). Therefore, an average number of retransmissions of

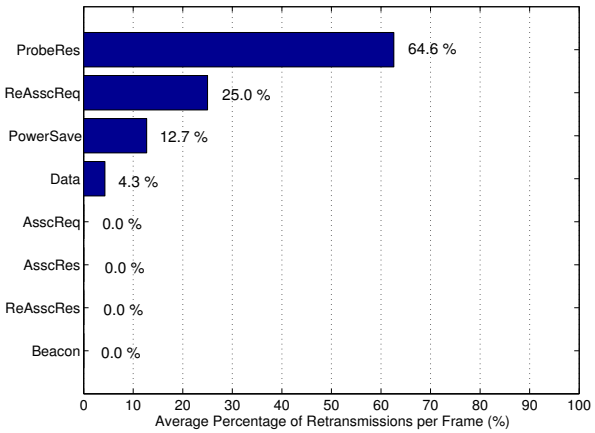

Fig. 22. [MAC Frame Types] Average number of retransmissions per MAC Type.

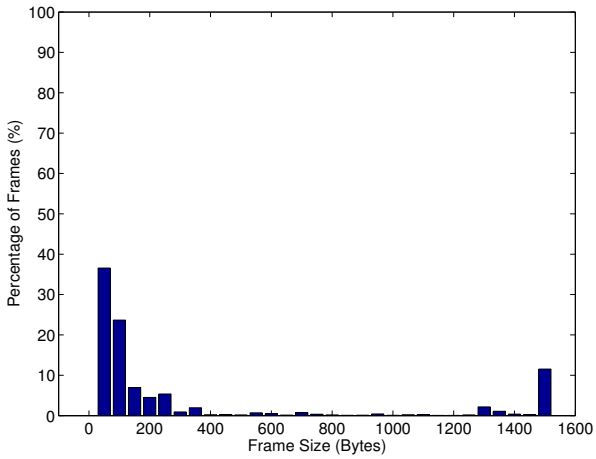

Fig. 23. [MAC Frame Size] Distribution of frame size (**Data** traffic only).

one indicates that each frame is retransmitted one time on the average.

We find that unexpectedly, ProbeRes, ReasscReq and PowerSave frames have a very high number of retransmissions on the average.

We give the following possible explanations for each case:

- *ProbeRes*: According to the 802.11 standard, a client sends a probe request on a channel, with a broadcast destination address, and waits on the same channel waiting for probe responses from the APs up to a maximum time (defined by the *MaxChannelTime* parameter). If there are multiple APs on the same channel, these APs will contend for the channel. This means that some APs will enter the backoff procedure. During this period, the client switches to another channel to scan for other APs. An AP returning from the backoff procedure will continue to send probe response frames, for a client who already left the channel not acknowledging the receipts of these frames, till it reaches the maximum number of retransmissions.
- *ReasscReq*: Although the client sends a request with a low data rate (indicating a poor signal condition (as shown from Fig. 31), the standard does not force the implementation to respond with a specific data rate. The AP sees from the ReasscReq that the client can support up to 11 Mbps and sends the ReasscRes with that rate[5]. Unfortunately, this message does not reach the client due to the poor signal conditions at the client side. This can be confirmed in the average data rate per frame types in Fig. 21.

  Another possible reason for this case is that the new AP needs to contact the old AP [22] to get the information before it can send the association response frame. This is a time consuming process during which the client time outs (as defined by the *ReassociateFailureTimeout* parameter in the 802.11 standard) and retransmits the reassociation request frame. This can explain why the AsscReq frames are not retransmitted although the corresponding AsscRes frames are sent at 11 Mbps. In this case, the AP does not need to contact any other AP and base its decision on a local policy.
- *Power-Save Poll*: When a STA wakes up, it sends a Power-Save Poll message to the AP asking for the buffered frames. The AP may have its *NAV* (Network Allocation Vector) set indicating that the medium is busy, so it cannot response to the poll message. Therefore, the station does not get a reply and retransmits the Power-Save Poll.

[5]The ReasscRes frame acts as the Acknowledgement in this case.

We believe that these high average retransmission for this frame types represents anomalies in either the protocol design or implementation. We are currently discussing these findings with a major 802.11 wireless card manufacturer.

*4) **MAC Frame Size**:* We can obtain the MAC frame size from the MAC header. In this section, we investigate the following questions: first, how MAC frame sizes are distributed; Is there any difference between the distributions for From-AP traffic and for To-AP traffic? Second, how much the MAC frame size affects the transmission performance.

To answer the first question, we plot the histogram of the frame sizes, based on the number of frames that have a certain size, in Figures 23, 24 and 25. The y-axis indicates the frequency of a certain frame size on the x-axis. We can observe that Data traffic has a bimodal shape, i.e. very small frames and very large frames are both frequently observed. In Fig. 24, we observe that distribution for From-AP Data traffic looks similar to that of aggregate Data frame, but has less small-size frames observed. In contrast, To-AP traffic has mostly small frames and a very low frequency of large frames. This shape is due to the request/response interactions between clients and the AP. Response traffic from the AP contains usually very big size frames, in order to transmit images and files. Since the MTU (Maximum Transmission Unit) size on the wired interface is 1500 Bytes, the maximum frame size of the wireless medium

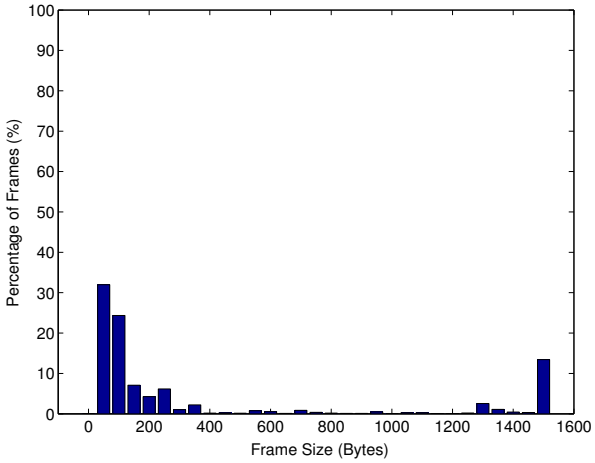

Fig. 24. [MAC Frame Size] Distribution of frame size (**From-AP Data** traffic only).

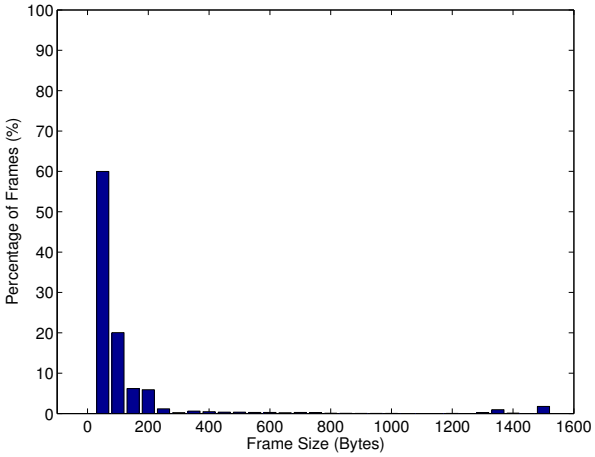

Fig. 25. [MAC Frame Size] Distribution of frame size (**To-AP Data** traffic only).

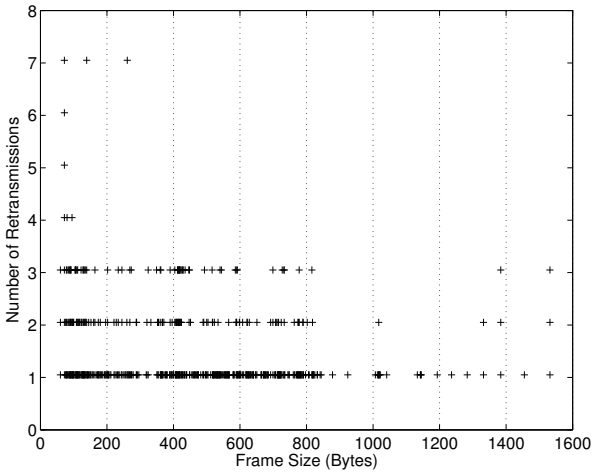

Fig. 26. [MAC Frame Size] Correlation between the number of retransmissions and frame size when **No RTS/CTS** is used (**To-AP** traffic on Friday, Feb. 13).

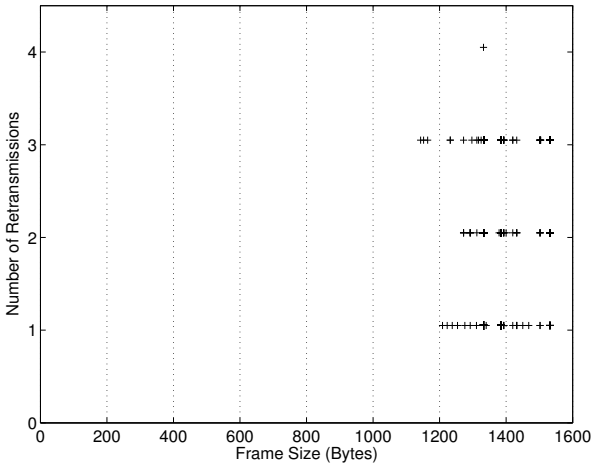

Fig. 27. [MAC Frame Size] Correlation between the number of retransmissions and frame size when **RTS/CTS** is used (for **From-AP** traffic on Friday, Feb. 13). Note that each point can represent multiple data points.

for From-AP traffic is 1500. That's why the From-AP traffic has high frequency of 1500-byte frames. Note also that for To-AP frames, we do not observe any frames with more than 1500 bytes. This strongly indicates that the wireless devices are not configured to use the MTU of the 802.11 protocol (2312 bytes). We believe that the reason for that is to avoid fragmentation at wired side whose MTU is 1500 bytes.

Fig. 26 shows the correlation between number of retransmitted frames observed and the corresponding frame sizes when RTS/CTS mechanism is **not** used for the To-AP case[6]. In the figures, each point represents a distinct frame observed, whose size and whose number of retransmissions are x and y coordinates respectively. When **no** RTS/CTS mechanism is used, the To-AP traffic experiences many transmissions errors (up to 7 retransmissions [7]). Note also that one may think that small-size frames have a higher number of retransmissions compared to large-size frames. However, Fig. 25 shows us that more than 90% of the frames have a frame size less than 200 bytes. This explains the high density of retransmissions at the low values of the x-axis in Fig. 26.

On the other hand, as shown in Fig. 27 and Fig. 28, RTS/CTS mechanism[8] reduces significantly the number of retransmissions. The density of the points near the high values of the frame size in the From-AP case and low values in the To-AP case can be explained by the frame size distribution in figures 24 and 25.

*5) **PHY Layer**:* We can obtain PHY layer information, such as data rate and signal strength from the Prism2 header. In this section, we investigate the distribution of data rate and correlation between data rate and signal strength. Some

[6]For the From-AP case, the access point always uses RTS/CTS for large frames.

[7]Retry limit of our target AP is set to 32.

[8]We correlate the RTS/CTS frames with the nearest data frame.

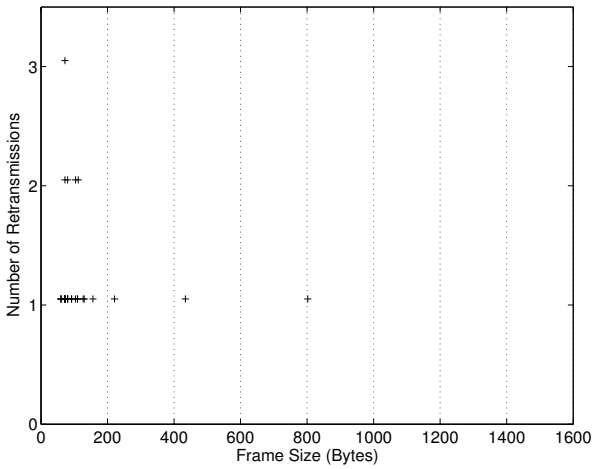

Fig. 28. [MAC Frame Size] Correlation between the number of retransmissions and frame size when **RTS/CTS** is used (for **To-AP** traffic on Friday, Feb. 13). Note that each point can represent multiple data points.

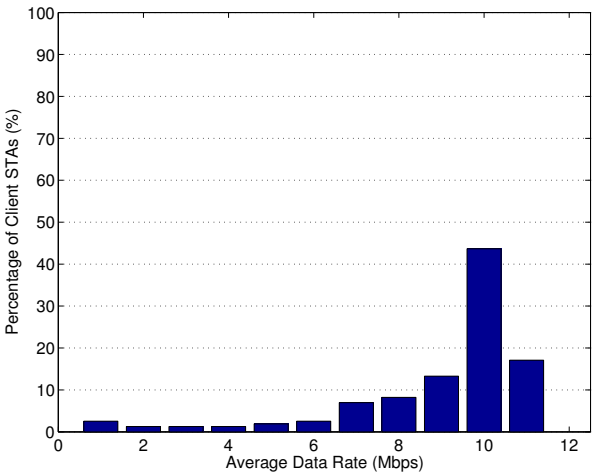

Fig. 30. [PHY (Data Rate)] Distribution of percentage of clients per average data rate (**To-AP Data** traffic only).

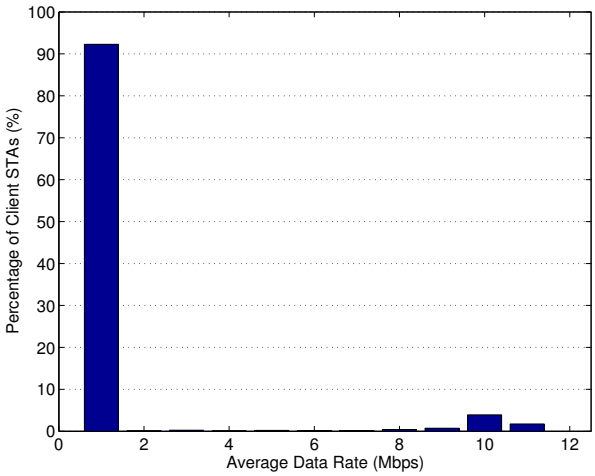

Fig. 29. [PHY (Data Rate)] Distribution of percentage of clients per average data rate (**From-AP Data** traffic only).

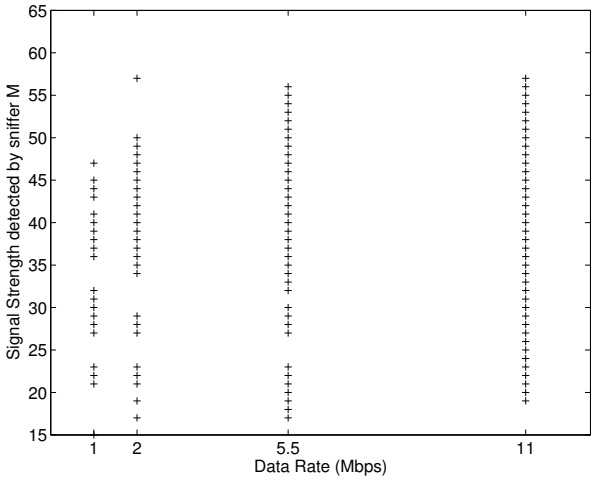

Fig. 31. [PHY (Signal Strength)] Correlation between signal strength and data rate (**To-AP Data** traffic only, captured by sniffer *T*).

cards, e.g. Lucent, run data rate adaptive algorithm, e.g. ARF (Auto-Rate Fallback) [23] where the card reduces the data rate to enhance the SNR. In this section, we are interested in observing such adaptations in our actual traffic data.

Fig. 29 shows how many clients use a certain range of data rates in From-AP traffic. We can observe that in From-AP traffic, the AP sends the frames to most of the clients with the lowest data rate. In contrast, in Fig. 30, we observe that the clients send the frames to the AP with relatively high data rates. One should expect that the lower the signal strength, at a client, the lower the data rate should be to enhance the SNR.

In Fig. 31, we obtain the signal strength detected by sniffer *T*, which can be assumed to be close to the signal strengths detected by the AP, because sniffer *T* is adjacent to the AP. We can see from the figure that there is no correlation between the signal strength detected by the AP and the data rate the client uses to send the frames. Put in another way, most clients do not adapt their data rate to compensate for bad signal conditions between them and the AP.

V. 802.11 Protocol Anomalies

Our study discovered several anomalies:

- IEEE 802.11 fragmentation mechanism is rarely used, if at all, in actual traffic. In our traffic measurement on the target AP, we did not observe any frames with fragmentation bit set to 1. We believe that using the fragmentation mechanism of the 802.11 protocol would reduce the number of retransmissions (especially for From-AP traffic, Fig. 27).
- Some management frames, e.g. association response and reassociation response frames, are transmitted at the highest data rate which does not correspond to the

client SNR conditions (Fig. 21). This leads to excessive retransmissions of these management frames.

- We observe significant number of retransmissions of the IEEE 802.11 Management frames. Those frames include Probe Response (64%), Reassociation Request (25%) and Power-Save Poll (13%). These retransmissions lead to the unnecessary waste of the scarce wireless capacity. We believe the reason for such retransmissions to be the incomplete specification of current MAC protocol. To prevent such anomalies, MAC protocol standards need to specify in more detail the frame exchange sequences and need to consider various conditions on PHY layer, e.g. data rate, signal strength, etc.
- Most of the clients fail to adapt the data rate according to the signal condition between them and the AP (Fig. 31). As a result, clients always use high data rate with poor signal conditions, which causes more transmission errors.

VI. Conclusions

In this paper, we introduced wireless monitoring as a technique to better characterize the wireless traffic. We showed that depending only on SNMP statistics and/or wired monitoring misses a lot of details about the operation of the wireless medium.

Using wireless monitoring, we get full access to the wireless frames including physical and MAC layer information which are not available using other analysis techniques. This allows us to get per client statistics for the wireless medium and to better analyze the wireless traffic. Moreover, we could identify anomalies of the operation of the 802.11 protocol.

However, wireless sniffing has the challenge of reduced capture performance due to the noisy characteristics of the wireless channel. We showed that every wireless sniffer has a different view of the wireless medium. We presented the multiple sniffer merging technique and sniffer placement strategy for increasing the wireless monitoring capture performance. Our results show that using these two techniques our wireless monitoring technique captures 99.34% of the wireless traffic, achieving capture performance comparable to the SNMP statistics and the wired monitoring technique.

We showed the results of using the wireless monitoring technique to analyze the traffic of an AP in a computer science department environment. Our MAC layer analysis showed the typical traffic mix of data and management frames, and their temporal characteristics and correlation with the user activities and the error characteristics of the wireless medium. Moreover, we showed the typical frame sizes and how the frame size affects the error rate. For the physical layer analysis, we showed the histogram of the data rates and how the data rate correlates with signal strength. Our results show that unexpectedly, the signal strength for the To-AP traffic is not correlated with the data rate, indicating that most clients do not use the data rate adaptive algorithm. Moreover, a large fraction of management frames are unnecessarily retransmitted leading to decreased capacity. We also showed that some protocol features included to enhance performance, like fragmentation, are rarely used.

Currently, we are working on extending our work in different directions. One direction is to scale the experiment to analyze the traces for multiple APs. This would give us information about the roaming pattern for users and how different APs on overlapping channels affect each other. In addition, we can use this analysis to provide models for multiple AP interaction. In such experiments, we expect that combining wired measurement with wireless monitoring would give better analysis capabilities. For example, wired monitoring can be used to analyze the *Inter Access Point Protocol* information [15]. Combining this information with the wireless monitoring analysis, we can study the roaming behavior of the mobile users and the handoff process. Extending the analysis to study other aspects of the 802.11 protocol is another direction being investigated. For example, the timing characteristics of the 802.11 protocol can be studied from the wireless traces. We can obtain traffic models by characterizing the inter-arrival time between MAC frames for different clients. Furthermore, the distribution of the time that each station spends in doze mode can also be estimated.

We believe that our results represent the first complete analysis of an 802.11b environment. Our sniffer merging technique and placement strategy can be used as a basis for larger experiments. In addition, our results can be used to build better models and simulators for the 802.11 protocol and the identified anomalies may be used by protocol designers and implementers for newer versions of the protocol.

References

[1] IEEE Computer Society LAN MAN Standards Committee. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. In *IEEE Std 802.11-1999*, 1999.

[2] A. Balachandran, G.M. Voelker, P. Bahl and V. Rangan. Characterizing User Behavior and Network Performance in a Public Wireless LAN In *Proc. ACM SIGMETRICS 2002*, Marina Del Rey, CA, June 2002.

[3] D. Kotz and K. Essien. Analysis of a Campus-wide Wireless Network. In *Proc. the Eighth Annual International Conference on Mobile Computing and Networking (MOBICOM 2002)*, Atlanta, GA, September 2002.

[4] D. Tang and M. Baker Analysis of a Local-Area Wireless Network In *Proc. the Sixth Annual International Conference on Mobile Computing and Networking (MOBICOM 2000)*, Boston, MA, August 2000.

[5] B.J. Bennington and C.R. Bartel. Wireless Andrew: Experience building a high speed, campus-wide wireless data network. In *Proceedings of MOBICOM*, September 1997.

[6] D. Eckardt and P. Steenkiste. Measurement and Analysis of the Error Characteristics of an In-Building Wireless Network. In *Proceedings of SIGCOMM*, August 1996.

[7] D. Schwab and R Bunt. Characterising the Use of a Campus Wireless Network In *Proc. IEEE INFOCOM 2004*, Hong Kong, China, March 2004.

[8] Y.C. Tay and K.C. Chua An Capacity Analysis for the IEEE 802.11 MAC Protocol. In *Wireless Networks*, January 2001.

[9] J. Bianchi Performance Aanlysis of the IEEE 802.11 Distributed Coordination Function. In *IEEE Journal On Selected Areas in Communications*, March 2000.

[10] T.S. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, 2002.

[11] J. Yeo, S. Banerjee and A. Agrawala Measuring traffic on the wireless medium: experience and pitfalls. Technical Report, CS-TR 4421, Department of Computer Science, University of Maryland, College Park, December 2002.

[12] J. Wright. Layer 2 Analysis of WLAN Discovery Applications for Intrusion Detection http://home.jwu.edu/jwright/papers/l2-wlan-ids.pdf
[13] S. Banerjee and A. Agrawala. Estimating Available Capacity of a Network Connection. In *Proceedings of IEEE International Conference on Networks*, September 2001.
[14] Y. Xiao, J. Rosdahl. Troughput and Delay Limits of IEEE 802.11. In *IEEE Communications Letters*, Vol. 6, No. 8, pp. 355-357, 2002.
[15] IEEE. Draft 5 Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation. *IEEE Draft 802.1f/D5*, January 2003.
[16] K. McCloghrie and M. Rose. RFC 1066 - Management Information Base for Network Management of TCP/IP-based Internets. TWG, August 1988.
[17] K. McCloghrie and M. Rose. RFC 1213 - Management Information Base for Network Management of TCP/IP-based Internets: MIB-II. TWG, March 1991.
[18] F. Baker and R. Coltun. RFC 2665 - Definitions of Managed Objects for the Ethernet-like Interface Types Network Working Group, August 1999.
[19] IEEE Computer Society LAN MAN Standards Committee. IEEE 802.11 Management Information Base In *IEEE Std 802.11-1999*, 1999.
[20] Centry Tap: Full-Duplex 10/100 Ethernet Splitter http://www.shomiti.net/shomiti/century-tap.html
[21] THE IMAP Connection http://www.imap.org/
[22] M. Shin, A. Mishra, and W. Arbaugh Improving the Latency of 802.11 Hand-offs using Neighbor Graphs. In *Infocom 2004*, Hong Kong, China, March 2004.
[23] A. Kamerman, L. Monteban WaveLAN-II: A High Performance Wireless LAN for the Unlicensed Band. Bell Labs Technical Journal, Vol. 2, No. 3, pp. 118-113, Summer 1997.
[24] Computer Associates. Virus Information Center (Win32.Netsky.B Virus) http://www3.ca.com/virusinfo/virus.aspx?ID=38332

APPENDIX I
WIRELESS MONITORING TECHNIQUE ON WIRELESS LAN TRAFFIC

In this section we describe our wireless monitoring technique and its effectiveness in terms of measurment loss and precise statistics for wireless traffic. Wireless monitoring technique with only one sniffer has severe drawback of high measurement loss, which also has high variability [11]. We can reduce such high measurement loss by placing multiple sniffers to capture the wireless traffics at the same time. However, there are several issues about this multiple sniffer technique: First, *how many* sniffers should we place at target location? Second, *where* should we place the sniffers? Third, how can we synchronize and merge the multiple IEEE 802.11 traffic traces, which are captured by multiple sniffers.

The first issue is out of scope of this paper, we are only concerned with the second and third issues. We propose and develop two techniques, namely *sniffer placement* and *merging multiple sniffers* for the second and third issues respectively. In this section we first describe our system setup for wireless monitoring in Section I-A, then discuss our merging method in Section I-B. Finally we explain where to place multiple sniffers in Section I-C.

A. System Setup

1) Measurement Hardware/Software: We set up several sniffer machines to capture the wireless frames on the air. All sniffing devices use Linux operating systems with kernel version 2.4.19. We used *Ethereal* (version 0.9.6) and *libpcap* library (version 0.7) with the *orinoco_cs* driver (version 0.11b) as sniffing software. We made use of the 'monitor mode' of the card to capture the IEEE 802.11 header as well as physical layer header, called the *Prism2* monitor header.

2) Captured Data : The sniffer captures the first 256 bytes of each receiving 802.11 frame, records the complete view of the frame, i.e. PHY/MAC/LLC/IP/Above-IP information.

Prism2 monitor header is not a part of IEEE 802.11 frame header, but is generated by the firmware of the receiving card. The header includes useful PHY information, such as MAC Time, RSSI(Received Signal Strength Indication), SQ (Signal Quality), Signal strength, Noise (in dBm) and Signal Noise Ratio (SNR) (in dB) and Data rate (in Mbps). We modify the orinoco_cs driver source codes to capture error statistics of the device at the time of capturing the frame. The error statistics include number of RX packets, number of frame errors and their reasons (e.g. CRC error, oversized/undersized frame, FIFO errors, etc.) and number of discarded frames.

We also capture the IEEE 802.11 MAC frame structure which incorporates the following fields: protocol version, frame type (management, data and control), Duration for Network Allocation Vector (NAV) calculation, BSS Id, Source and Destination MAC addresses, fragment, sequence number etc [1]. According to the 802.11 MAC frame type of the captured frame, we extract different information. For example, for Beacon frames, captured information include 64-bit Beacon timestamp which we use for time synchronization among multiple sniffers (refer to Section I-B.1 for more details). For Association/Dissassociate and Authentication/Deauthentication frames, the information include the reason code for such actions.

For the above-MAC layer, we first examine LLC (Logical-Link Control) type. If the type is IP, then we extract IP information such as IP Identification, IP source address, IP destination address, IP protocol type (e.g. UDP/TCP). On UDP/TCP packets, we record source port and destination port for application-level statistics. We record TCP sequence number and acknowledge number for TCP-specific statistics.

B. Merging Multiple Traces

In this section, we describe our merging technique, specifically method of time synchronization and merging procedures respectively.

1) Time Synchronization between Multiple Traces : To merge the multiple IEEE 802.11 MAC traffic traces, we need to synchronize the timestamp of each trace in significantly high resolution, i.e. at least in tens of microseconds. Those timestamps are measured on different machines with different wireless devices. We want to synchronize the sniffers within the same BSS (Basic Service Set), therefore all the sniffers are assumed to associate the same AP.

To correctly distinguish the IEEE 802.11 frames, we require the time synchronization error (the difference between two timestamps of different sniffers for the same frame) to be less than the minimum gap between two valid IEEE 802.11 frames. In the IEEE 802.11b, the minimum gap G_{min} can be calculated by 192 microsecond preamble delay plus 10

microsecond SIFS (Short Inter-Frame Space), therefore to be 202 microsecond. In the IEEE 802.11a, the minimum gap G_{min} can be calculated by 20 microsecond preamble delay plus 4 microsecond symbol delay plus 16 microsecond SIFS, therefore to be 40 microsecond [14].

Although in this paper we apply our technique only in the IEEE 802.11b wireless networks, we require the synchronization error to be less than $40\mu s$, so that our technique can be applied to any current IEEE 802.11 standards. With this synchronization error requirement, we can correctly identify the same frame, therefore can remove the duplicate frames in multiple sniffer traces.

a) Synchronization with Reference Timestamp: We use linear regression to fit one timestamp to another, therefore we need *a priori* common frames among all the sniffers. We choose the IEEE 802.11 Beacon frames, which are generated by the AP, to be the common frames to all the sniffers. The Beacon frames contain their own 64-bit absolute timestamps, therefore we can uniquely identify such common beacon frames in different snffer traces. Assume we have three different sniffers S1, S2 and S3. To precisely represent the receiving time of one common Beacon frame, we use the MAC time of receiving frame, which is available in prism2 header in the captured frame. Because we need the exact, i.e. high-resolution, time when frame reception occurs in sniffer device, we do not use the timestamps generated by sniffer's operating system.

We can have four different timestamps for one common Beacon frame: S1's MAC timestamp (T_{S1}), S2's MAC timestamp (T_{S2}), S3's MAC timestamp (T_{S3}) and Beacon's own timestamp (T_B). Now we want to fit the timestamps of three sniffers, i.e. target timestamps, to the *refererence timestamp* using linear function. In other words, we need to find a linear function to convert the target timestamps to the reference timestamp. In our setup, reference timestamp T_r can be either T_B, T_1, T_2 or T_3. We assume that any taget timestamp T_s at sniffer s can be linearly converted to the predicted timestamp $\tau_r^{(s)}$, which is compatible with the reference timestamp T_r as follows:

$$\begin{aligned} \tau_r^{(s)} &= \beta_r^{(s)} T_s + \alpha_r^{(s)}, \\ R_r(s) &= \tau_r^{(s)} - T_r, \end{aligned}$$

where $\alpha_r^{(s)}$ and $\beta_r^{(s)}$ are constants and $R_r(s)$ is called a *residue* at sniffer s, defined to be the difference, i.e. fitting error, between predicted timestamp $\tau_r^{(s)}$ and the actual reference timestamp T_r.

To evaluate each synchronization method, we can also define $SE(s_1, s_2)$, the synchronization error between sniffer s_1 and s_2 on a common Beacon frame as follows:

$$SE(s_1, s_2) = |\tau_r^{(s_1)} - \tau_r^{(s_2)}|.$$

We need to determine the values of α_r and β_r using *Least Square Method* on all the receiving common Beacon frames during the *fitting interval*. Fitting interval is the valid range of a fitting function and it is important to choose a fitting interval for the fitting performance and computing overhead. For example, we can fit the target timestamps to reference timestamps only during the interval of consecutive common beacons (usually 100 ms). In this case fitting performance, i.e. in terms of small synchronization error, is the best, however we should find as many fitting functions as the number of consecutive common beacons. If we choose a fitting interval during 20 common beacons or more, then the fitting performance is worse than the interval of consecutive two common beacons, but we can save a lot of computing overhead.

TABLE V
TIMING RELATION BETWEEN T_s AND T_B : EXPERIMENTAL EVIDENCE

Data rate (before)	Data rate (after)	Interval bet'n T_B	Interval bet'n T_s	Interval bet'n $T_B - T_{TX}$
1 Mbps	1 Mbps	102260	102259	102260
1 Mbps	5.5 Mbps	**103937**	**104016**	**104006**
5.5 Mbps	5.5 Mbps	102457	102456	102457
5.5 Mbps	2 Mbps	**122871**	**122840**	**122844**
2 Mbps	2 Mbps	92958	92956	92958

According to which timestamp we can choose as T_r, in our setup we have the following four options of synchronizatoin method :

$$\begin{aligned} \text{REF\_B: } \tau_B &= \beta_B T_s + \alpha_B, \\ \text{REF\_S1: } \tau_{S1} &= \beta_{S1} T_s + \alpha_{S1}, \\ \text{REF\_S2: } \tau_{S2} &= \beta_{S2} T_s + \alpha_{S2}, \\ \text{REF\_S3: } \tau_{S3} &= \beta_{S3} T_s + \alpha_{S3} \end{aligned}$$

According the IEEE 802.11 standard [1], the Beacon timestamp should equal to the time when the timestamp is generated by the AP plus the transmission delay of the frame, therefore T_B reflects the time when the sniffer receives the *last* bit of a Beacon frame irrespective of data rate. Therefore, we need to confirm whether T_s at each sniffer s is generated at the reception of the *first* bit of a Beacon frame or at the reception of the *last* bit of the frame.

To confirm this timing relation between T_s and T_B, we perform the above four synchronization methods on 13-minute wireless traces captured by three sniffers, namely *kif*, *zapp* and *mclure*. We examine the trace captured by *kif*, especially the part where the change in the data rate of Beacon frames occurs. In Table V, we compare the interval between consecutive T_B, the interval between consecutive T_{kif} and the interval between consecutive $T_B - T_{TX}$ where T_{TX} is the transmission delay of Beacon frame in a given data rate. We show the case of no change in data rate in row 1, 3 and 5. In the second and the fourth rows, the data rate of Beacom frames increases (from 1Mbps to 5.5 Mbps) and decreases (from 5.5 Mbps to 2 Mbps) respectively.

In Table V, we can confirm that T_s at sniffer s is generated when s receives the *first* bit of the Beacon frame. When data rate changes, e.g. in the second and fourth row in Table V, the interval between T_s approximately equals to the interval

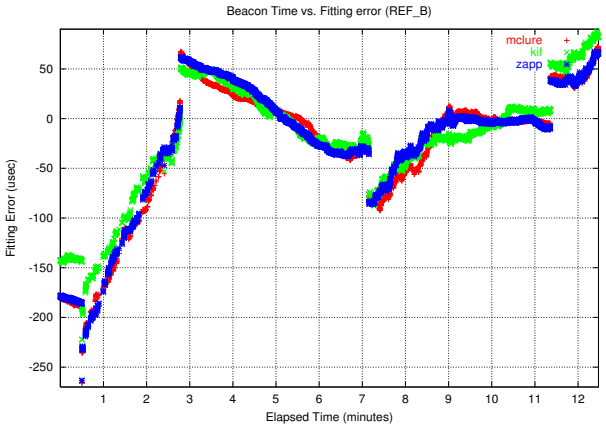

Fig. 32. Residue (fitting error) $R_B(zapp)$, $R_B(kif)$ and $R_B(mclure)$ in REF_B method with 5658 common beacon interval. Zapp, kif and mclure indicates the three wireless sniffers.

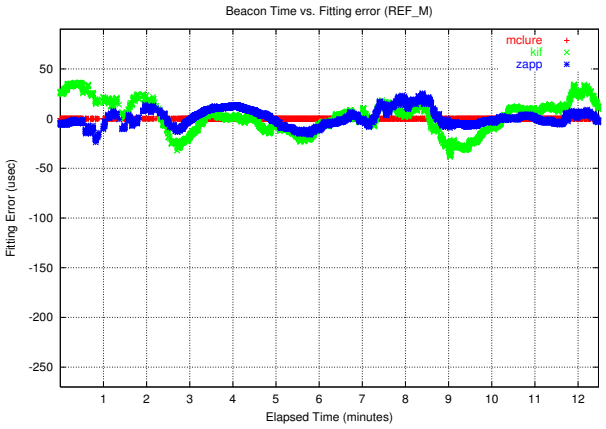

Fig. 33. Residue (fitting error) $R_M(zapp)$, $R_M(kif)$ and $R_M(mclure)$ in REF_M (REF_Mclure) method with 5658 common beacon interval.

between $T_B - T_{TX}$, where T_{TX} is the transmission delay of Beacon frame in a given data rate. We can also observe that when data rate increases, the interval between T_B's becomes less than the interval between T_s's (in the second row in Table V) and when data rate decreases, the interval between T_B's becomes greater than that between T_s's (in the fourth row). Therefore we can confirm that T_B equals to the time of sniffer receiving the *last* bit of the Beacon frame transmitted by the AP.

Based on this timing relation between T_s and T_B, we can notice the problem of REF_B method, that is, whenever the data rate of Beacon frames changes, the AP resets T_B. Put in other way, whenever the data rate changes, the clock T_B becomes a new clock, therefore is very unstable in terms of the usage of reference time.

To identify this problem emperically, we examine the residue R_B, i.e. difference between predicted timestamp of each sniffer and T_B, in Figure 32. We observe that many discontinuities occur in Figure 32 and the times of those discontinuites coincide with when data rate of Beacon frames changes due to variable channel condition. On the other hand, in REF_Mclure (REF_M, in short) method in Figure 33, such discontinuities are not noticed.

Therefore, to correctly convert the sniffer timestamp to Beacon timestamp, we need to use the adjusted time, $T_B - T_{TX}$, instead of using T_B, so that the Beacon timestamp can be stable even when the data rate changes. We need to use the following equation instead of REF_B equation:

$$\text{ADJ\_B: } \tau_A = \tau_B - T_{TX} = \beta_A T_s + \alpha_A.$$

In the followings, we compare the results of the above four synchronization methods applied on the real data, ADJ_B, REF_M, REF_K (REF_kif) and REF_Z (REF_zapp) in terms of synchronization error $SE(s_1, s_2)$. We also compare various fitting intervals (e.g. interval of two common Beacons, 200 common Beacons, 1000 common Beacons and more) to determine which can best satisfy our requirement of synchronization error.

b) Performance of Time Synchronization Methods: On three 13-minute traces captured by three sniffers, *mclure*, *kif* and *zapp*, we perform the four synchronization methods to obtain the synchronization error $SE(s_1, s_2)$ among the three sniffers.

We find that all the three sniffers observe 5658 common Beacons from an AP (say, AP1) during 13 minutes. We generate the synchronization parameters $\alpha_r^{(s)}$ and $\beta_r^{(s)}$ for each sniffer MAC timestamp T_s for each synchronization interval. For example, if we apply ADJ_B method with the interval of 1000 common Beacons to the 13-minute trace, then we have 6 $(= \lceil 5658/1000 \rceil)$ intervals, each of which has distinct fitting equation. For the trace of *mclure*, we calculate 6 pairs of $\alpha_B^{(mclure)}$ and $\beta_B^{(mclure)}$ for each interval from T_{mclure} and $T_B - T_{TX}$ of the common Beacon frames. In the same way, we also calculate those parameters for the traces of *kif* and *zapp*. We use Matlab function *robustfit()* to calculate the synchronization parameters α and β.

Once we have the synchronization parameters for each sniffer for each synchronization interval, we apply those parameters to the sample data to obtain $SE(M, K)$, $SE(K, Z)$ and $SE(Z, M)$ respectively. The sample data consists of 5681 common Beacons, which however are from another AP (say, AP2). Therefore, we can compare the synchronization performance on different sample data from the 5658 common Beacons we use for generating synchronization parameters.

In Table VI, we present the *maximum* values of $SE(s_1, s_2)$, i.e. the maximum difference between predicted timestamps of two sniffer s_1 and s_2, of different synchronization methods with various synchronization intervals (10000, 1000, 200 and two in our experiment). We can observe that as the interval decreases, the synchronization performance increases, i.e. maximum SE decreases. The method ADJ_B with the interval of 5658 performs worse than other methods. However, with the interval of 1000 or shorter, all the four methods show good performances, i.e. they satisfy the error requirement: less than 40 microseconds.

TABLE VI

PERFORMANCE OF SYNCHORONIZATION METHODS WITH VARIOUS FITTING INTERVALS: THE INTERVAL IS IN UNIT OF NUMBER OF COMMON BEACONS. $SE(s_1, s_2)$ IS IN MICROSECOND. M, K AND Z INDICATE MCLURE, KIF AND ZAPP, WHICH ARE THE NAMES OF THE THREE WIRELESS SNIFFERS.

Method	Maximum Synchronization Error $SE(s_1, s_2)$ between two sniffers s_1 and s_2											
	interval of 5658			interval of 1000			interval of 200			interval of two		
	(M,K)	(K,Z)	(Z,M)	(M,K)	(K,Z)	(Z,M)	(M,K)	(K,Z)	(Z,M)	(M,K)	(K,Z)	(Z,M)
ADJ_B	74	82	26	27	19	20	17	17	15	4	4	4
REF_M	39	44	25	26	20	20	19	23	26	3	3	3
REF_K	39	59	38	26	20	20	19	20	23	3	3	3
REF_Z	52	59	25	26	20	20	15	20	26	3	3	3

To synchoronize long-duration wireless traces, we need to confirm that our technique guarantees the error requirement on the traces of much longer than 13 minutes. We merge three 24-hour long traces which are simultaneously captured by three sniffers *mclure*, *kif* and *zapp* on June 6, 2003. As a result of applying REF_Z method with the interval of two common Beacons, the average SE between *merged* frames, i.e. the frames estimated to be the same frames, is 1.4 microsecond. We also obtain 31 microsecond of maximum SE between merged frames, which satisfy the error requirement. We will describe the merging procedures in detail in the following section.

2) Merging Procedures: We assume that there are N traces denoted by $s_1, s_2, \ldots s_N$, which are captured by N sniffers, durations of which should be overlapped for some period of time (we use $s_1, s_2, \ldots s_N$ to represent the names of the sniffers or the traces, interchangeably without ambiguity). We assume that from the same frames captured by different sniffers at the same time, we can extract the exact the same information to identify their equality. We can use any of the four synchronization methods for this merging procedure, but assume we use REF_r, where r is one of N sniffers, with the interval of M_i common Beacons. Therefore, the total number, say M_c, of common Beacons observed by N sniffer is greater than or equal to M_i.

- *Step 1*: Generate a *signature* file from each trace so that we have N signature files from N traces. A signature file consists of as many *signatures* as the number of captured frames in the trace. A signature of a captured 802.11 frame is an one-line summary of the frame, which consists of at maximum 56 fields, seperated by whitespaces. The number and semantics of fields follow the descriptions in Section I-A.2.
- *Step 2*: Scan N signature files at the same time, one-pass with each file, to find M_c common Beacons. From the M_c common Beacons, generate one *SYNC* file. The SYNC file consists of $\lceil M_c/M_i \rceil$ lines, each of which corresponds to each synchronization interval. Therefore, each line contains the corresponding fitting parameters: N number of each sniffer s's MAC timestamp for the *starting* common Beacon frames during the interval and N pairs of $\alpha_r^{(s)}$ and $\beta_r^{(s)}$, where s denotes the name of each sniffer.
- *Step 3*: Merge incrementally N signature files by first merging the files for s_1 and s_2 into s_{12}, then merging s_{12} and s_3 into s_{123}, and so on (therefore, in total $N-1$ steps for N signature files). We only describe the procedure for merging s_1 and s_2, without loss of generality. As we read each frame from the signature file s_1 (s_2), we identify the synchronization interval by comparing T_{s_1} (T_{s_2}) of the frame with the T_{s_1} (T_{s_2}) of the starting common Beacon in the SYNC file. Once we identify the corresponding interval, we convert T_{s_1} (T_{s_2}) to $\tau_r^{(s_1)}$ ($\tau_r^{(s_2)}$) so that we can compare the two timestamps in reference sniffer s_r's clock time. Now suppose we read s_1's k'th frame f_{1k} and we denote its converted timestamp to be $\tau_r^{(s_1)}(f_{1k})$. Then we construct the *comparison window* of f_{1k}, whose range is $[\tau_r^{(s_1)}(f_{1k}) - G_{min}, \tau_r^{(s_1)}(f_{1k}) + G_{min}]$. Here, G_{min} is the minimum gap between two valid IEEE 802.11 frames, e.g. 202 microsecond in the IEEE 802.11b, as described in Section I-B.1. Then, we compare the contents of f_{1k} with only the frames of s_2 whose converted timestamps lie within the comparison window. If the contents of the two compared frames are the same, then we remove the matched s_2 frame from the window. After the comparison, we flush, i.e. write on the merged file, s_2's frames whose timestamps are before the window of f_{1k}, i.e. before $\tau_r^{(s_1)}(f_{1k}) - G_{min}$. Then we flush the s_2's frames in the window whose timestamps are before f_{1k}, then flush f_{1k}. Finally we flush the remaining s_2's frames in the window while they are *not* included in the $(k+1)$'th window of the next frame $f_{1(k+1)}$. We repeat this procedure until we encounter the end of trace s_1 or s_2. If we encounter the end of trace s_1 then flush the remaining frames in s_2 to the end, and vice versa.

C. Placement of Monitoring Devices

In this section, we describe our strategy for determining the locations of multiple sniffers. We first measure Signal-Noise Ration of the signals from APs, then determine the coverage of a specific AP based on the SNR values.

1) Measurement of SNR (Signal-Noise Ratio) : We measure the SNR of the signals from the AP (SNR from the AP, in short) using wireless monitoring technique, i.e. at some measurement location we capture any Beacon frames from

any AP observed there and examine the SNR values in Prism2 header of the frames. In this way, we measure the SNRs at as many locations as possible in the target area, then from the measurement data we extract only the SNR from the target AP to find the marginal SNR line, therefore we can determine the coverage of the target AP, which is defined in the following section.

In Figure 34, we show the 15-dB marginal line, i.e. the coverage of the AP, determined by applying the measurement technique. We give ESSID "mindlab" to the AP, therefore we setup an one AP wireless network. In other words, there are no other APs nearby with the same ESSID. At 132 measurement locations in the fourth floor of A.V. Williams building, we capture Beacon frames to obtain the average SNR for 30 seconds (therefore about 300 Beacon frames per AP) at each location. Then in off-line we extract the average SNRs from the target AP (with ESSID "mindlab") to obtain the SNR contour lines of 40, 30, 20 and 15 dB, as shown in Figure 34.

We can measure the SNR from the target AP by associating the measurement device with the AP and examining the SNR only from the AP. However, the problem of this method is that it is difficult to force the measurement device to be associated with the AP, especially at the location far from the AP.

Applying wireless monitoring technique to measure the SNR gives another advantage that we can also obtain the SNRs from the other APs on same or other channels. Based on the SNRs from any observable APs at the location, we can obtain the coverage of the APs in terms of the definition in previous section in case there are other APs with the same ESSID as the target AP. We obtain the coverages of ESSID "umd" APs in the fourth floor of A.V. Williams building. In their coverages, the APs have greater SNR than from other APs, as shown in Figure 6.

2) Placement Strategies Based on SNR Measurement: Since in the infrastructure mode of the 802.11 protocol all traffic goes through the AP, one may think that placing all sniffers near the AP should maximize the capture performance. However, our experiments showed that the capture performance of To-AP traffic is worse than that of the From-AP traffic, even for the sniffer T which was adjacent to the AP. This is due to the weak signal that reaches the sniffer from the clients compared to the strong signal that reaches the same sniffer from the AP. The AP can capture the weak signal due to its better hardware and specialized processing (compared to the sniffer configuration).

Therefore, for placing the wireless sniffers, we should only place *one sniffer* adjacent to the AP to be responsible for capturing the From-AP traffic and the traffic of clients near the AP. Other sniffers should be placed as close as possible to the wireless clients.

If we assume that clients are going to be uniformly distributed over the coverage area, this translates to placing the sniffers so that they cover as much as possible from the AP coverage area. Therefore, if we have n sniffers to place, we can split the AP coverage area into n equal areas and place the sniffers in the center of mass of these areas.

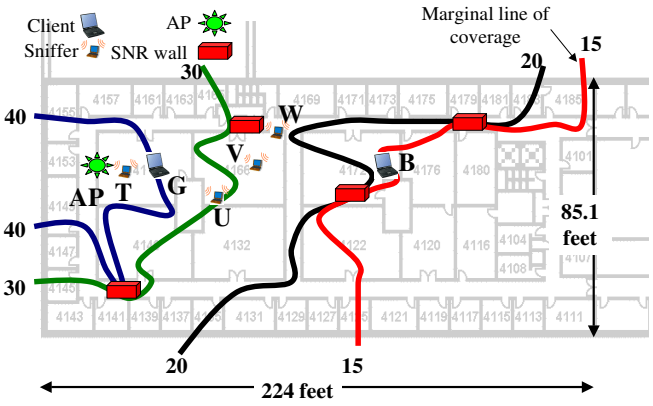

Fig. 34. SNR Contour Map for controlled experiment: SNR Contour lines for 40,30,20 and 15 dB are obtained from SNR measurements. Based on the contour map, we place the wireless stations at location G and B, place the sniffers at location T, U, V and W. Several *SNR Walls* are detected by examining the distances between adjacent contour lines.

We can refine this strategy by noting that, in an environment where multiple APs are installed, the coverage area of an AP may be reduced to the *Association Area* of the AP. The *Association Area* of an AP is the area at which a client will favor this AP for association compared with other APs in the area. Note that the Association area is a sub-area of the coverage area and that most of the traffic an AP receives comes from the associated clients (i.e. from the Association Area). Therefore, we should use the association area of an AP rather than its coverage area. Fig. 6 shows the Association Areas for different access points in the area of interest. The figure also shows the difference between the coverage area and the association area for AP_1

Another factor that needs to be taken into account is the signal condition at the sniffer location. We define an *SNR wall* as an area where the SNR contour lines are close to each other (Fig. 34). Our experiments (see Section II-D) shows that placing a sniffer near an SNR wall leads to worse capture performance compared to placing the sniffer at other places. Therefore, SNR walls should be avoided.

APPENDIX II
CONTROLLED EXPERIMENTS

To examine the effect of our wireless monitoring techniques, i.e. *merging multiple sniffers* and *sniffer placement*, we conduct controlled experiments using *NetDyn*, which we will describe in the following section.

The purpose of this experiment is two-fold: first we examine the effect of our monitoring techniques, i.e. multiple sniffer merging and placement, and second we quantitatively compare the effect of our wireless monitoring technique with those of the other measurements, e.g. end-to-end measurement, SNMP and wired monitoring. The analysis in terms of the second purpose will be discussed in Section **??**. We describe the setup of the controlled experiments in this section.

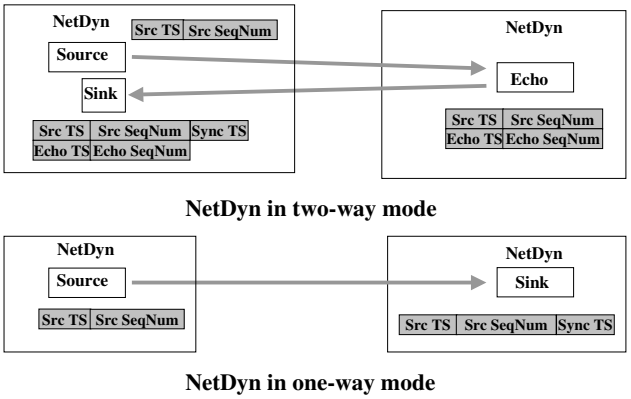

Fig. 35. Controlled Experiment using NetDyn: *Source* in a wireless station sends 20000 UDP packets to wired *Echo* machine which sends them back to *Sink* in the same wireless station.

TABLE VII

CONTROLLED EXPERIMENT SETUP: MODE INDICATES NETDYN MODE, EITHER ONE-WAY OR TWO-WAY.G DENOTES THE STATION AT LOCATION G, B DENOTES THE STATION AT LOCATION B, S DENOTES THE NETDYN SERVER, WIRED BEHIND THE AP.

Exp. #	*Mode*	*Src.*	*Dst.*	*# Pkts*	*Inter-Pkt Time*
Exp. 1	Two-way	G	S	20000	10 ms
	Two-way	B	S	20000	10 ms
Exp. 2	One-way	B	S	20000	5 ms
	One-way	S	G	20000	5 ms
Exp. 3	One-way	S	B	20000	5 ms
	One-way	G	S	20000	5 ms

A. Methodology

1) Environment : We perform our experiments in the fourth floor of A.V. Williams building at University of Maryland (where Department of Computer Science is located). The building has 58 access points installed, which belong to three different wireless networks. Each wireless network is identified with its *ESSID*. The ESSIDs of the three networks are *umd*, *cswireless* and *nist* respectively. *umd* network consists of 29 Cisco Aironet A-340 APs, and is the most widely used wireless network in the university. *cswireless* (12 Lucent APs) and *nist* (17 Prism2-based APs) are built by individual research research groups in the department.

2) NetDyn: End-to-end Traffic Measurement Tool: NetDyn [13] is an end-to-end traffic measurement tool, in which UDP or TCP packets are actively injected to the network for measuring *available bandwidth* of the network.

NetDyn consists of three different processes, *Source*, *Echo* and *Sink* and can operate in two modes, namely *one-way* or *two-way*. In two-way mode, *Source* process on sender machine generates a packet with timestamp and source packet number and sends it to to *Echo* process on remote machine. As the *Echo* process receives the packet, it attaches its timestamp and echo packet number to the packet, then send the packet back to *Sink* process, which is running also on sender machine. After *Source* sends some number of packets and *Sink* receives them back, we can calculate the number of packets correctly transmitted between *Source* and *Echo* by comparing the source packet number and echo packet number of the packets successfully received by *Sink* process. We can also calculate RTT (Round Trip Time) between *Source* machine and *Echo* machine.

In one-way mode, *Source* process resides on the sender machine, and *Sink* process is running on the receiver machine. As the name implies, the traffic is only from *Source* process to *Sink* process. In one-way mode, we can calculate only the number of packets successfully transmitted from *Source* to *Sink*, but cannot obtain RTT between them.

This tool was originally designed for measuring the available bandwidth in wide-area network, we use it in this work for generating and measuring the traffic in wireless LAN. Fig. 2 shows the setup for NetDyn running in two-way mode on wireless LAN. *Source* and *Sink* processes run on a wireless station (a mobile laptop) in our setup, while *Echo* process runs on a server wired to Ethernet Local Area Network. By *Source*, *Sink* and *Echo*, we mean in this paper those processes or the machines where they run.

3) Setup for Wireless Network: Because we need more controlled environment to examine the effectiveness of our technique quantitatively, we set up a separate wireless network with ESSID "mindlab" in this experiment.

We have one Orinoco AP-1000 wireless access point with Orinoco Silver wireless card at the location *AP* in Figure 34. The AP is wired to the server machine S, on which we run NetDyn processes, *Source*, *Sink* and *Echo*. We have two wireless stations, namely G and B at locations *G* and *B* in Figure 34, on each of which NetDyn processes *Source* and *Sink* run. Location *G* and *B* represents a *Good* location in terms of signal condition and a *Bad* location respectively. *G* is located at 40-dB line 12-feet apart from the AP and *B* is on the 15-dB *marginal* line of SNR from the AP. The two wireless stations are equipped with the same wireless card as the AP.

4) Setup for Wireless Monitoring, Wired Monitoring and SNMP: We place the three sniffers at locations *T*, *U* and *V* respectively in Figure 34. Those sniffers run the same H/W and S/W described in Section I-A. We merge the three traces with REF_V (*V* is also used for the name of the sniffer) with the interval of two consecutive common Beacons from the AP.

To measure the *wired* traffic, we set up one *wired* sniffer between the server S and the AP. The *wired* traffics between the server S and the AP in both direction are captured on the wired sniffer through the "Century Tap", a full-duplex 10/100 Ethernet splitter [20]. We capture the wired traffics using *Ethereal* on the wired sniffer S.

On the wired sniffer, we also run a SNMP client, *snmputil*, which is written by Balachandran *etal.* for the previous research [2]. We make *snmputil* polling the AP every minute, similar to the setup in [2].

B. Generating Wireless Traffics

For generating traffic as realistic in actual wireless LAN as possible, we run three different experiments, the setup of which are summarized in Table VII. In Experiment 1 (Exp. 1, in short), we have two *two-way* NetDyn traffics between two wireless stations G, B and wired server S. Wireless station G (B) runs *Source* and *Sink* processes, at the same time the server S runs *Echo*. G (B) send 20000 packets of full UDP payloads (1472 bytes) to S, with 10 ms inter-packet. In other direction, S sends back the packets to G and B. Therefore in Exp. 1, we have generated UDP traffic of roughly 4.7 *Mbit per second* (aggregately in both directions) on wireless medium.

In Exp. 2, we instead have two *one-way* NetDyn traffics between two wireless stations G, B and wired server S. Wireless station B runs *Source* process to send 20000 packets of full UDP payloads (1472 bytes) with 5 ms to corresponding *Sink* process on the server S At the same time in other direction, the *Source* process on the server S sends the same amount of traffic to the *Sink* on another wireless statin G. Therefore in Exp. 2, we have also generated UDP traffic of roughly 4.7 *Mbit per second* on wireless medium. Exp. 3 has the same setup as Exp. 2 except having the traffics in reverse directions.

Exp. 1 emulates the realistic situation that a STA in good location in terms of signal condition and another STA in bad location both try to communicate with the remote server(s). Exp. 2 (Exp. 3) is a scenario similar to the situation that one sender STA in bad (good) location and one receiver STA in good (bad) location run at the same tme in the same wireless LAN.

C. The Effect of Merging

We estimate the measurement loss improvement of merging multiple sniffers by conducting an controlled experiment. We can calculate the measurement loss by looking at MAC sequence number field as shown in [11], but we find that this method is not accurate due to the misbehaviors of various wireless cards and the AP. We notice that there exist some (in one trace 41 frames out of 125057) *out-of-sequence* frames in their MAC sequence number. By out-of-sequence frames, we mean one frame's MAC sequence number is less than the previous frame's MAC sequence number even though the interval between two frames is not long enough for the two numbers to wrap around. The range of the IEEE 802.11 MAC sequence number is $[0, 4095]$, therefore for two sequence numbers of consecutive frames to wrap around, the interval should be greater than $G_{min} * 4096$.

Others reported that such out-of-sequence MAC sequence number can be generated by some wireless stations for malicious attack [12]. Therefore, for estimating the exact measurement loss, we need to use reliable sequence numbers generated by the applications. This is the reason why we conduct two-way UDP packet exchange experiments using an end-to-end traffic measurement tool, called *NetDyn* [13], described in Section II-A.2.

TABLE IX

THE EFFECT OF PLACEMENT (WITH THREE SNIFFERS AT LOCATIONS T, U AND V): THE NUMBER OF NETDYN PACKETS CAPTURED ON EACH SNIFFER IS PRESENTED. THE COLUMN *Dist.* SHOWS THE NUMBER OF DISTINCT PACKETS, THE COLUMN *All* ALSO INCLUDES THE NUMBER OF RETRANSMISSIONS. THE NUMBERS ARE NORMALIZED WITH RESPECT TO THE NUMBER CAPTURED BY SNIFFER AT T.

Traffic	*Sniffer at T*		*Sniffer at U*		*Sniffer at V*	
	Dist.	*All*	*Dist.*	*All*	*Dist.*	*All*
Exp.1 (G→S)	100	100	96.0	95.8	95.2	95.1
Exp.1 (G←S)	100	100	98.9	98.8	96.8	96.7
Exp.1 (B→S)	100	100	143.3	171.5	143.5	172.5
Exp.1 (B←S)	100	100	99.8	129.9	100.2	128.0
Exp.2 (B→S)	100	100	185.7	217.5	186.6	219.7
Exp.2 (G←S)	100	100	99.4	99.4	98.7	98.6
Exp.3 (G→S)	100	100	89.9	89.8	89.0	89.0
Exp.3 (B←S)	100	100	100.5	101.3	99.9	98.3
Total (G↔S)	100	100	96.4	96.3	95.3	95.2
Total (B↔S)	100	100	126.5	145.2	126.7	144.3
Total (From-AP)	100	100	99.6	107.1	98.8	105.2
Total (To-AP)	100	100	121.2	134.4	120.9	134.6

Table VIII shows the effect of using the merged sniffers' traces. We show the result of Exp. 1 only, but in other experiments we have similar observations. We can see from the table that increasing the number of merged sniffers' traces from one to two to three increases the percentage of captured frames significantly from 73.25% to 84.47% to 99.34% respectively for the To-AP traffic. Notice also that the effect of merging is more significant in the case of To-AP traffic while a single sniffer near the AP (sniffer T) can almost capture all the From-AP traffic (improvement from T only to T+U+V is 0.7%). Using the merged three-sniffers ' trace, wireless monitoring can capture more than 99.34% of the wireless traffic.

D. The Effect of Placement

To examine the effect of our placement technique described in Section I-C, we place three sniffers according to the technique during the controlled experiment in Table VII.

In Figure 34, for capturing the traffic between the AP and the STA at location G (STA G, in short), we place one sniffer at the middle point between them, i.e. at location *T*. As another STA is at location B (STA B, in short), which is on the marginal line of the AP coverage, we place the other two sniffers around the middlie point between the STA and the AP.

For the placement of the other two sniffers we choose three points *U*, *V* and *W*. *U* is located exactly on 30-dB line, *V* is closer to the client STA B than 30-dB line. *W* is more off than 30-dB line, even closer to STA B than *V*. Noticeably, the location *W* is close to one of the *SNR Wall*s, which is detected by the SNR contour lines in Figure 34.

Due to only three sniffers available at the same time, we run the set of experiments in Table VII two times. In the first round, we place the sniffers at *U* and *V* and run Exp 1., Exp.2 and Exp.3. In next round of Exp. 1-3, we place the sniffers at *U* and *W* and conduct the same experiments.

TABLE VIII

INCREASING CAPTURED FRAMES BY MERGING MULTIPLE SNIFFERS: MERGING TWO OR THREE SNIFFERS AMONG T, U AND V SIGNIFICANTLY INCREASES THE NUMBER OF OBSERVED FRAMES. (WE SHOW THE RESULT OF EXP. 1 ONLY)

	To-AP Wireless Traffic							
	NetDyn	*T*	*U*	*V*	*T+U*	*T+V*	*U+V*	*T+U+V*
Good	19905	76.76%	69.00%	68.34%	76.83%	70.00 %	76.84%	98.61%
Bad	18490	69.48%	99.58%	99.73%	99.05%	100.0 5%	99.97%	100.13%
Total	**38395**	**73.25%**	**83.73%**	**83.46%**	**87.54%**	**84.47%**	**87.98%**	***99.3 4%***
	From-AP Wireless Traffic							
Good	19247	98.41%	97.31%	95.24%	99.37%	98.06%	99.32%	99.38%
Bad	17858	102.04%	101.85%	102.2%	102.56%	102.43%	102.52%	102.56%
Total	**37105**	**100.15%**	**99.5%**	**98.59%**	**100.91%**	**100.16%**	**100.86%**	***100.91%***

TABLE X

THE EFFECT OF PLACEMENT (WITH THREE SNIFFERS AT LOCATIONS T, U AND W)

	Sniffer at T		*Sniffer at U*		*Sniffer at W*	
Traffic	*Dist.*	*All*	*Dist.*	*All*	*Dist.*	*All*
Exp.1 (G→S)	100	100	87.3	87.2	84.5	84.4
Exp.1 (G←S)	100	100	96.5	96.7	87.6	87.7
Exp.1 (B→S)	100	100	138.7	171.0	**136.3**	**169.2**
Exp.1 (B←S)	100	100	100.2	100.2	98.3	93.2
Exp.2 (B→S)	100	100	125.7	146.7	**125.4**	**147.7**
Exp.2 (G←S)	100	100	93.4	93.3	72.9	72.8
Exp.3 (G→S)	100	100	80.4	80.3	78.4	78.4
Exp.3 (B←S)	100	100	100.0	96.7	97.5	88.1
Total (G↔S)	100	100	89.5	89.5	81.2	81.1
Total (B↔S)	100	100	115.1	122.5	113.4	117.8
Total (From-AP)	100	100	97.4	97.1	89.0	86.4
Total (To-AP)	100	100	105.4	117.5	103.5	116.0

In Table IX, we show the number of NetDyn packets captured by the three sniffers (denoted by sniffer *T*, *U* and *V* respectively) in the first round of experiments. We observe that in both distinct number and all number (retransmissions included), the sniffer *T* performs best for the traffic between STA G and the server S (therefore between G and the AP). This is due to the fact that sniffer at *T* is the closest to both the AP and STA G. Likewise, for G-S traffic, the sniffer *U* performs better than the sniffer *V* due to its physically closeness to G.

For the traffic between STA B and the server S, therefore the AP, the sniffer *U* captures more $B \leftarrow S$ than the sniffer *V*, while the sniffer *V* captures more $B \rightarrow S$ than the sniffer *U*. This also agrees with their relative closeness to S and B respectively.

Table X shows the number of captured frames by the sniffers when we place them at *T*, *U* and *W*. Even though the location *W* is much closer to STA B than the location *U*, the performance of the sniffer *W* for $B \rightarrow S$ is no better than the sniffer *U*, even slightly worse. We infer that the SNR Wall right next to the location *W* affects the performance of the sniffer *W*. For the traffics other than $B \rightarrow S$ traffic, the sniffer *W* shows worse performance than the sniffer *V*, which agrees with its physical remoteness from the AP.

From Table VIII, Table IX and Table X we conclude as follows:

- The location of the sniffer is one of the most influencing factors in sniffing performance in terms of number of captured packets.
- Different sniffers have different viewpoints of the wireless medium.
- The *absolute* physical location of the client or the sniffer does not affect the ability of a particular sniffer to capture data from a particular wireless client. Rather, the *relative* position between the wireless client and the sniffer is the factor that affects the ability of a sniffer to capture the data from that client.
  For example, for the traffic originating from STA B, sniffers *U* and *V* capture more traffic than sniffer *T*. The reason for that is as the distance from the sniffer to the wireless client increases, the signal strength decays and the SNR decreases leading to worse signal conditions and decreased sniffing performance. Sniffers *U* and *V* are closer to STA B than sniffer *T*.
- For From-AP traffics, showed in *Total (From-AP)* row, the sniffer *T* outperforms other sniffers in *Distinct* numbers, but with retransmissions included, other sniffers can perform better. This means that the sniffer dedicated to the AP reflects more the AP-oriented view, while others show more the STA-oriented or medium-oriented view.
- SNR Walls affect the sniffer performance, therefore we should avoid placing the sniffer near the SNR Walls even if the place agrees our placement strategy.

APPENDIX III
COMPARISON WITH END-TO-END MEASUREMENT, SNMP AND WIRED MONITORING

In the controlled experiment in the previous section, we have also measured the traffic with other methods, e.g. wired monitoring and SNMP. In this section, we *quantitatively* compare the results obtained from those methods with our wireless monitoring technique. We apply the techniques of *multiple sniffer merging* to obtain the statistics on wireless traffics with the experimental setup in Table VII. This comparison can

provide the *pros and cons* of our wireless monitoring technique compared to the other methods, prevalently used for wireless LAN measurement.

A. Comparison Results

As Table XI shows, we have seven statistics from various methods, e.g. end-to-end, SNMP based on MIB-I, SNMP based on MIB-II, wired monitoring and wireless monitoring. We also show in TableXII, XIII, XIV and XV, the statistics of the same experiments, but presenting per-experiment, per-station details.

We compare those methods with respect to the six values, # Packets, # ReTX (Retransmissions), # Errors in inbound (STA→server S) and outbound (S →STA) traffics.

To compare the various values from different measurement techniques, we should note that each statistics has different meaning. Therefore those statistics *cannot be blindly compared value by value* without careful consideration of the semantics. For example, # Errors in end-to-end measurement indicates # Frames not received by the destination application, while # Errors in SNMP measurement is defined to # Frames not sent (or received) by the AP. Therefore, we first explain in detail the semantics and the result of each statistics, then compare each statistics with others.

- *End-to-end measurement using NetDyn (Table XI-1)*: we obtain # NetDyn UDP packets correctly received by the destination by examining packet numbers generated by NetDyn processes. We define # Errors to be the number of NetDyn packets which are sent by NetDyn *Source* process but NOT received by NetDyn *Sink* process. Therefore this number can include the number of errors occuring at TX/RX buffers on *Source* or *Sink* machines, the errors on wired medium, the errors at the AP, and the errors on wireless medium. With the end-to-end trace at each STA and the server S, we can break down the statistics into the traffic from/to Good STA (at location G) and the traffic from/to Bad STA (at location B). In total, inbound traffic is more successfully transmitted than outbound traffic. In only inbound traffic, the traffic from Good STA (G, in short) experiences more errors than Bad STA (B). The reason is that the signal condition during Exp. 3 becomes much worse than during Exp. 2 (refer to Table VII for experimental setup). Therfore, during Exp. 3 the one-way experiment $G \rightarrow S$ experiences severe losses, while the one-way experiment $B \rightarrow S$ in Exp. 2 does not. On the other hand, in outbound traffics, the traffics to G has much more successful transmissions than those to B. This is partly due to the same reason describe above that the traffic $S \rightarrow B$ in Exp. 3 experiences worse signal condition. The another reason is due to the "bad" location of STA B.
- *SNMP measurement with MIB-I (Table XI-2)*: we retain the SNMP log of up-to-date aggregate statistics at the AP in one minute interval. From this log, we obtain the difference between the statistics at the start of the experiment and the statistics at the end of the experiment. Because the resolution of measurement time is one minute, the numbers presented can have maximum error up to the numbers of two-minute intervals (during about 30 minute experiments). According to the definitions in [16] and [2], we can define the statistics as follows: # Inbound Packets indicates # unicast frames successfully delivered to a higher-layer protocol (e.g. IP). # Inbound Errors indicates # errors preventing frames from being deliverable to a higher-layer protocol. # Outbound Packets is defined to be the total number of packets that higher-level protocols request to be transmitted to a subnetwork-unicast address. Note that # Outbound Packets include those frames that are discarded or not sent due to the errors. We calculate # Outbound ReTX using the technique in [2] by taking the difference between the number of inbound packets at *wired* interface and # Outbound Packets. # Outbound Errors is defined to be the number of outbound frames that could not be transmitted because of errors. Note that # Outbound Errors contains the number of errors in broadcast packets, e.g. Beacons. Intuitively, # Inbound Packets is the number of frames the AP receives successfully, # Outbound Packets is the number of the AP sends, and both are the exact statistics the AP can measure. However, # Errors is the aggregate number of errors in both unicast and broadcast packets, therefore the exact error statistics for only unicast packets are not available with MIB-I. In sub-table 2, we notice 8.2% of inbound errors and 3.8% of outbound errors, which are both higher than 2.41% error rate, which are obtained with the same method in public conference [2]. In their setup they have four different APs, but within the similar area we have only one AP, therefore worse signal conditions of our setup result in higher error rate. In their results, the number of (outbound) retransmissions is less than the number of errors, and they explain the mismatch with the errors in Beacon frames. In contrast, even though the same method is applied in our results, # Outbound ReTX is about 5 times more than # Outbound Errors. This implies that in our controlled experiment the location of STA B incurs severe retransmissions between B and the AP. # ReTX is the value estimated using the method in [2]. We will show more accurate statistics on errors and retransmission using MIB-II in the followings, then compare the MIB-II measurement result with the result of our wireless monitoring technique.
- *SNMP measurement with MIB-II (Table XI-3)*: we can obtain more accurate statistics with MIB-II [17], [18]. Especially we can have accurate statistics on errors and *outbound* retransmissions for unicast packets. MIB-II transmission group defines various objects for dot3 type interface, i.e. ethernet-like interface type [18]. For outbound retransmission, the variables *SingleCollisionFrames*, *MultipleCollisionFrames* and *ExcessiveCollisionFrames* are defined. *SingleCollisionFrames* and *MultipleCollisionFrames* indicate the number of frames retransmitted once and two times respectively to be finally *successfully*

TABLE XI

QUANTITATIVE COMPARISON BETWEEN END-TO-END MEASUREMENT USING NETDYN, SNMP, WIRED MONITORING AND WIRELESS MONITORING

	Inbound (To-AP) Wireless Traffic			*Outbound (From-AP) Wireless Traffic*		
Statistics	# Packets	# ReTX (%)	# Errors (%)	# Packets	# ReTX (%)	# Errors (%)
1. NetDyn End-to-End Measurement						
# Packets: # DISTINCT NetDyn UDP packets correctly received by the destination,						
# Errors: # DISTINCT NetDyn UDP packets NOT received by the destination.						
1. Total	75310	-	4690 (5.9%)	69403	-	9102 (11.4%)
(Good STA)	36831	-	3169 (7.9%)	39247	-	658 (1.6%)
(Bad STA)	38479	-	1521 (3.8%)	30156	-	8444 (21.1%)
2. SNMP Measurement with MIB-I in [2]						
# Inbound Packets: # frames received by AP without errors/discards.						
# Inbound Errors: # errors preventing frames from being deliverable to a higher-layer protocol.						
# Outbound Packets: # frames TXed by AP, including errors/discards.						
# Outbound ReTX: the difference between # inbound packets at wired interface and # outbound packets at w ireless interface.						
# Outbound Errors: # errors at AP, including the errors in broadcast packets.						
2. Total	75494	-	6727 (8.2%)	71959	14536 (16.8%)	2740 (3.8%)
3. SNMP Measurement with MIB-II						
# Packets: defined same as MIB-I.						
# Inbound Errors: sum of variables FCSErrors and InternalMacReceiveErrors.						
# Outbound ReTX: sum of variables SingleCollisionFrames, MultipleCollisionFrames and ExcessiveCollisions.						
# Outbound Errors: sum of variables ExcessiveCollisions, LateCollisions, CarrierSenseErrors and InternalMacTransmitErrors.						
3-1. Total	75494	-	6726 (8.2%)	71959	10793 (15.0%)	2944 (3.8%)
# Packets: # successfully RXed/TXed frames.						
# Outbound ReTX: # successfully ReTXed frames.						
3-2. Total	75494	-	-	69015	7849 (11.4%)	-
4. Wired Monitoring						
# Packets: # DISTINCT frames found in the wired traces.						
4. Total	75312	-	-	78335	-	-
(Good STA)	36832	-	-	39908	-	-
(Bad STA)	38480	-	-	38427	-	-
5. Wireless Monitoring checked with the IEEE 802.11 ACK						
# Packets: # DISTINCT observed data frames with the corresponding 802.11 ACK observed,						
# Errors: # DISTINCT observed data frames without 802.11 ACK observed,						
(32.4%) (34.2%)						
5. Total	53475	-	19956 (27.2%)	50183	-	21198 (29.7%)
(Good STA)	24421	-	10512 (31.8%)	28958	-	10150 (32.4%)
(Bad STA)	29054	-	9444 (24.5%)	21225	-	11048 (34.2%)
6. Wireless Monitoring checked with NetDyn results						
# Packets: # DISTINCT observed data frames found to be in NetDyn traces,						
# Errors: # DISTINCT data frames NOT observed but found to be in NetDyn traces.						
6. Total	73364	-	1940 (2.6%)	69236	-	167 (0.2%)
(Good STA)	34912	-	1913 (5.2%)	39092	-	155 (0.4%)
(Bad STA)	38452	-	27 (0.1%)	30144	-	12 (0.0%)
7. Wireless Monitoring checked with Retransmissions						
)						
# Packets: # DISTINCT observed data frames,						
# Errors: # DISTINCT observed retransmissions, i.e. frames with 802.11 RETRY field set.						
7. Total	73431	11815 (16.1%)	-	71381	9980 (14.0%)	-
(Good STA)	34933	1170 (3.3%)	-	39108	394 (1.0%)	-
(Bad STA)	38498	10645 (27.7%)	-	32273	9586 (29.7%)	-

TABLE XII

NETDYN RESULTS, PER EXP, PER STA: ONLY DISTINCT FRAMES ARE COUNTED.

	To-AP			*From-AP*		
Traffic	*# Packets*	*# ReTX*	*# Err. (%)*	*# Packets*	*# ReTX*	*# Err. (%)*
Exp.1 (G→S)	19905	-	95 (0.5%)	-	-	-
Exp.1 (G←S)	-	-	-	19247	-	658 (3.3%)
Exp.1 (B→S)	18490	-	1510 (7.6%)	-	-	-
Exp.1 (B←S)	-	-	-	17858	-	632 (3.4%)
Exp.2 (B→S)	19989	-	11 (0.1%)	-	-	-
Exp.2 (G←S)	-	-	-	20000	-	0 (0.0%)
Exp.3 (G→S)	16926	-	3074 (18.5%)	-	-	-
Exp.3 (B←S)	-	-	-	12298	-	7812 (39.1%)
Total (G↔S)	36831	-	3169 (7.9%)	39247	-	658 (1.6%)
Total (B↔S)	38479	-	1521 (3.8%)	30156	-	8444 (21.1%)
Total	75310	-	4690 (5.9%)	69403	-	9102 (11.4%)

TABLE XIII

WIRELESS MONITORING CHECKED WITH ACK, PER EXP, PER STA: ONLY DISTINCT FRAMES ARE COUNTED.

	To-AP			*From-AP*		
Traffic	*# Packets*	*# ReTX*	*# Err. (%)*	*# Packets*	*# ReTX*	*# Err. (%)*
Exp.1 (G→S)	13905	-	5725 (29.2%)	-	-	-
Exp.1 (G←S)	-	-	-	13522	-	5607 (29.3%)
Exp.1 (B→S)	13437	-	5078 (27.4%)	-	-	-
Exp.1 (B←S)	-	-	-	12741	-	5575 (30.4%)
Exp.2 (B→S)	15617	-	4366 (21.8%)	-	-	-
Exp.2 (G←S)	-	-	-	15436	-	4543 (22.7%)
Exp.3 (G→S)	10516	-	4787 (31.3%)	-	-	-
Exp.3 (B←S)	-	-	-	8484	-	5473 (39.2%)
Total (G↔S)	24421	-	10512 (31.8%)	28958	-	10150 (32.4%)
Total (B↔S)	29054	-	9444 (24.5%)	21225	-	11048 (34.2%)
Total	53475	-	19956 (27.2%)	50183	-	21198 (29.7%)

TABLE XIV

WIRELESS MONITORING CHECKED WITH NETDYN, PER EXP, PER STA: ONLY DISTINCT FRAMES ARE COUNTED.

	To-AP			*From-AP*		
Traffic	*# Packets*	*# ReTX*	*# Err. (%)*	*# Packets*	*# ReTX*	*# Err. (%)*
Exp.1 (G→S)	19619	-	289 (1.5%)	-	-	-
Exp.1 (G←S)	-	-	-	19113	-	134 (0.7%)
Exp.1 (B→S)	18480	-	10 (0.0%)	-	-	-
Exp.1 (B←S)	-	-	-	17854	-	4 (0.0%)
Exp.2 (B→S)	19972	-	17 (0.0%)	-	-	-
Exp.2 (G←S)	-	-	-	19979	-	21 (0.1%)
Exp.3 (G→S)	15302	-	1624 (9.6%)	-	-	-
Exp.3 (B←S)	-	-	-	12290	-	8 (0.0%)
Total (G↔S)	34912	-	1913 (5.2%)	39092	-	155 (0.4%)
Total (B↔S)	38452	-	27 (0.1%)	30144	-	12 (0.0%)
Total	73364	-	1940 (2.6%)	69236	-	167 (0.2%)

TABLE XV

WIRELESS MONITORING CHECKED WITH RETRANSMISSIONS, PER EXP, PER STA: ONLY DISTINCT FRAMES ARE COUNTED.

	To-AP			*From-AP*		
Traffic	*# Packets*	*# ReTX (%)*	*# Err.*	*# Packets*	*# ReTX (%)*	*# Err.*
Exp.1 (G→S)	19630	576 (2.9%)	-	-	-	-
Exp.1 (G←S)	-	-	-	19129	386 (2.0%)	-
Exp.1 (B→S)	18515	5121 (27.7%)	-	-	-	-
Exp.1 (B←S)	-	-	-	18316	4181 (22.8%)	-
Exp.2 (B→S)	19983	5522 (27.6%)	-	-	-	-
Exp.2 (G←S)	-	-	-	19979	8 (0.0%)	-
Exp.3 (G→S)	15303	594 (3.9%)	-	-	-	-
Exp.3 (B←S)	-	-	-	13957	5405 (38.7%)	-
Total (G↔S)	34933	1170 (3.3%)	-	39108	394 (1.0%)	-
Total (B↔S)	38498	10645 (27.7%)	-	32273	9586 (29.7%)	-
Total	73431	11815 (16.1%)	-	71381	9980 (14.0%)	-

transmitted. *ExcessiveCollisionFrames* is an error statistics for the number of frames retransmitted but *failing* to be transmitted. Other error statistics for outbound interface include *CarrierSenseErrors* i.e. # times that the carrier sense condition was lost, *LateCollisions* i.e. # times that collision is detected after 512 bit is transmitted and *InternalMacTransmitErrors* for other kinds of errors. The objects for inbound errors include *FCSErrors* i.e. the number of errors in Frame Check Sequence, and *InternalMacReceiveErrors* for other kinds of errors. With those variables, in sub-table 3-1 in Table XI, we have exact # Inbound Errors, # Outbound ReTX and # Outbound Errors. We can also calculate the statistics for only RX/TX successes by subtracting # Errors from # Packets, as shown in sub-table 3-2. Note that # Inbound Packets in sub-table 3-1 does not include # Errors, therefore has the same value in both (3.1) and (3.2).

- *Wired monitoring (Table XI-4)*: from the wired traces captured between the server S and the AP, we obtain DISTINCT number of packets to/from each STA. Most of the frames captured by wired sniffer are NetDyn packets (75333 out of 75340 for inbound traffic, 78348 out of 78355 for outbound traffic). Most of the frames captured are NOT duplicated. Only 28 (20) packets are duplicated for inbound (outbound) traffic. With source and destination IP addresses available in the traces, we can break down the statistics per STA. The number of outbound packets captured by wired monitoring, i.e. 78348, is much greater than actual end-to-end successful transmissions, i.e. 69403, by roughly 10000 packets. This shows that as the signal condition become worse, the accuracy of wired monitoring is severely reduced.
- *Wireless monitoring checked with the IEEE 802.11 ACK (Table XI-5)*: We calculate the statistics on the *merged* trace from three different wireless sniffers *T*, *U* and *V* in Figure 34. One way to obtain the statistics for successful transmissions is to examine the IEEE 802.11 Acknowledge (ACK, in short) frame for observed data frame. If we can observe ACK frame also, we can regard the data packet to be successfully transmitted. Otherwise, we can infer that either the transmission fails or ACK is not observed by any of the three sniffers. As sub-table 4 shows, significant amount of ACK frames are not observed by the sniffers, i.e. 27.2% on inbound traffic and 29.7% on outbound traffic. The numbers of successful transmissions also show significant difference from the numbers in end-to-end result in sub-table 1, SNMP result in sub-table 3-2 and wired monitoring in sub-table 4. This implies that we should not rely only on this statistics, need another statistics to obtain accurate traffic statistics from the wireless traces. One thing we may note is that our sniffers observe fairly equivalent number of frames from/to STA G and STA B. In other words, irrespective of end-to-end results, or whether the location of a STA is good or bad, our placement strategy does have fairness over the STA's in terms of the number of captured packets.
- *Wireless monitoring checked with NetDyn results (Table XI-6)*: What if we ignore the ACK frames, instead count any observed frames to be successfully transmissions? In sub-table 6 shows that this method produces the result very close to the result we can obtain in end-to-end measurement. Out of 75310 packets successfully transmitted on inbound end-to-end traffics, the sniffers observe 73357 packets, i.e. more than 97%. On outbound traffic, the sniffers also observe more than 99% of successfully transmitted end-to-end packets. The result in sub-table 6 has a significant improvement from the result in sub-table 5, however we still have a problem of so-called *false positive* packets. If we count all the observed frames to be the successful transmissions, we may have some packets, which the sniffers observe but is not actually successfully transmitted in end-to-end layer. We can use the distribution of retransmission counts

to distinguish such false positive packets from actual successful transmissions. We will discuss this issue later in this section.

- *Wireless monitoring checked with retransmissions (Table XI-7)*: One of most important advanvantages of wireless monitoring technique is to capture the statistics of retransmissions accurately both on inbound and outbound traffics. In sub-table 7, we notice that 16.1% of frames are retransmitted on inbound traffic, 14.0% of frames are retransmitted on outbound traffic. We can confirm the accuracy of this statistics by comparing this sub-table and sub-table 3-2 on outbound retransmissions. SNMP based on MIB-II records 10793 retransmissions, wireless monitoring captures 9980 of them. The error in accuracy is less than 8%. Most (more than 90%) of retransmissions are to/from STA B and the location of STA B is marginal, therefore we can think this 8 % error to be the upper bound.

As described above, we show that the accuracy and effectiveness of our technqiue by comparing # Packets, # ReTX and # Errors with other techniques. We can summarize our findings as follows.

- In SNMP techniques [2], [16], MIB-I method does not provide accurate statistics because it uses many estimations rather than actual recorded counters.
- MIB-II method provides more accurate statistics on # ReTX and # Errors, but due to its polling interval (e.g. one minute), the numbers can be deviated from the correct statistics by at most two polling intervals.
- With the statistics of MIB-II result to be the baseline, the wireless monitoring can capture more than 90% of outbound retransmissions. Our technique also provides the retransmissions on inbound traffic, which cannot be obtained by SNMP method.
- As compared with the end-to-end result, the wireless monitoring can observe more than 97% of successful transmissions at end-to-end application layer.
- In noisy environment, wired monitoring produces severely incorrect statistics on outbound traffic.

B. MAC-level Retransmissions

In Table XVI, we show the number of retransmissions on inbound and outbound traffics, whose maximum number is set to three. For example, out of 73431 *distinct* frames observed by the sniffers, 14% are retransmitted once, 2% are retransmitted twice and 0.2% are retransmitted three times (Maximum Retry is set to 3). Likewise out of 83494 frames (including duplicate frames), 74% are observed not retransmitted, 12% are retransmitted once, 4% are twice and very few are retransmitted three times. Sniffers observe 8288 frames (10% of all frames), which are the first transmitted frames, are observed to be retransmitted at least once.

These numbers are based on observations by sniffers, therefore we need to confirm that the numbers are close to the actual numbers of successful transmissions. For example, 61626 frames are observed to be transmitted without retransmissions.

TABLE XVI

MAC-LEVEL RETRANSMISSIONS OBSERVED BY SNIFFERS: *Dist.* INDICATES NUMBER OF DISTINCT FRAMES OBSERVED BY THE SNIFFERS. 'NO RETX' IS THE NUMBER OF FIRST TRANSMITTED PACKETS, WHOSE RETRANSMISSIONS ARE *not* OBSERVED. '1ST TX' IS THE NUMBER OF FIRST TRANSMITTED PACKETS, WHOSE RETRANSMISSIONS ARE OBSERVED.

	Inbound (To-AP)		*Outbound (From-AP)*	
# ReTX	*# Dist. (%)*	*# All (%)*	*# Dist. (%)*	*# All (%)*
No ReTX	61626 (84)	61626 (74)	61401 (86)	61401 (73)
1 ReTX	10196 (14)	10196 (12)	6353 (9)	6353 (7)
2 ReTX	1473 (2)	2946 (4)	2171 (3)	4342 (5)
3 ReTX	146 (0)	438 (0)	1456 (2)	4368 (5)
1st TX	0 (0)	8288 (10)	0 (0)	8222 (10)
Total	73431 (100)	83494 (100)	71381 (100)	84686 (100)

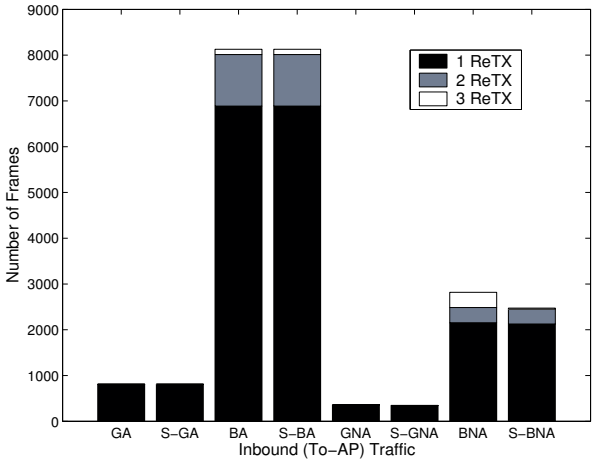

Fig. 36. Retransmissions in Inbound (To-AP) Traffic: GA indicates the frames from Good STA with ACK observed, S-GA indicates Su ccessfully TXed GA. GNA indicates the frames from Good STA without ACK observed. Likewise BA (BNA) indicates the frames from Bad ST A with (without) ACK observed.

How many frames out of them are actually successfully transmitted? If significant number of the frames fail to be transmitted, then we could not trust the retransmission statistics.

Figure 36 shows the distribution of observed retransmissions on inbound traffics and how many frames out of them are actually successfully transmitted. Figure 37 shows the same but on outbound traffics. We break down the retransmission statistics per good or bad STA, at the same time as to whether the IEEE 802.11 ACK frame is also observed or not. We find that most of the retransmissions of good STA are actually successfully transmitted. (All of the 'No ReTX' frames are successfully transmitted, which is not shown in Figure 36 and Figure 37. Also most of the retransmissions whose ACK frames are observed, are actually successfully transmitted.

The only traffics which make different the number of retransmissions and the number of successful transmissions out of them, are those of bad STA without ACK frames observed. Therefore, if we can calculate the number of successfully

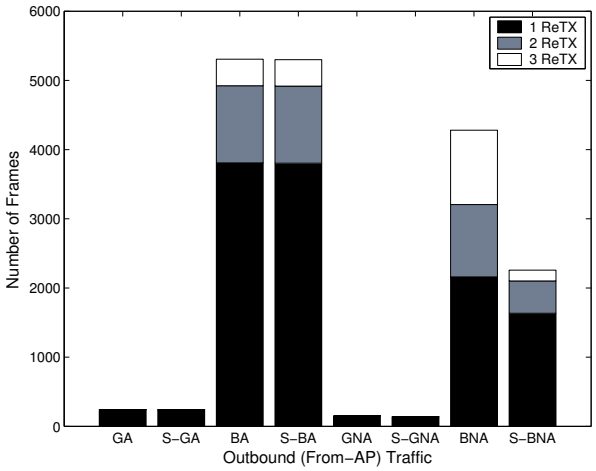

Fig. 37. Retransmissions in Outbound (From-AP) Traffic: GA indicates the frames to Good STA with ACK observed, S-GA indicates S uccessfully TXed GA. GNA indicates the frames to Good STA without ACK observed. Likewise BA (BNA) indicates the frames to Bad STA w ith (without) ACK observed.

TABLE XVII

ESTIMATION OF # SUCCESSFUL TRANSMISSIONS FROM # OBSERVED RETRANSMISSIONS: BNA INDICATES THE RETRANSMITTED FRAMES OF BAD STA WITHOUT ACK OBSERVED.

	Inbound BNA		*Outbound BNA*	
n ReTX	*Actual* s_n	*Estimated* s_n	*Actual* s_n	*Estimated* s_n
1	0.99	0.96	0.75	0.85
2	0.97	0.99	0.45	0.52
3	0.78	0.9	0.15	0

transmitted frames out of observed retransmitted frams whose ACK are not observed, then we can obtain the accurate statistics of successful transmissions.

How can we calculate the number of successful transmissions? Suppose we know p_1, the transmission failure probability of 1st retransmitted frames. Likewise, we suppose we know p_2 and p_3. Then s_1, the probability of the 1st retransmission being successfully transmitted ultimately is given as follows:

$$s_1 = (1-p_1) + p_1 * (1-p_2) + p_1 * p_2 * (1-p_3). \quad (1)$$

Likewise we can calculate s_2 and s_3 with the following equations:

$$s_2 = (1-p_2) + p_2 * (1-p_3), \quad (2)$$

$$s_3 = 1-p_3. \quad (3)$$

Therefore, once we know p_1, p_2 and p_3, we can obtain the number of successful transmissions from the number of retransmissions. Now the question is how we can estimate p_1 and p_2. We can estimate p_1 by the ratio of the number of 1st retransmissions and the number of 2nd retransmissions. Generally, we can estimate p_n by the ratio of the number of n'th retransmissions and the number of $(n+1)$'th retransmissions.

In Table XVII, we show that this estimation technique can calculate the numbers of successful transmissions close the actual numbers.