Assessing the Accuracy of COTS 802.11b/g Sniffers

Pablo Serrano
Departamento de Ingenieria de Telematica, Universidad Carlos III de Madrid,
Leganes, Madrid, Spain and
Michael Zink, Jim Kurose
Department of Computer Science, University of Massachusetts Amherst,
Amherst, MA 01003, USA

Abstract

Many recent measurement studies have analyzed WLAN performance by means of wireless sniffers. However, little attention has been given to the accuracy of sniffers themselves, as the common assumption is that the sole reason for frame losses is erroneous reception (e.g., bit-level errors) or collisions. In this paper, we experiment with three different wireless packet sniffers in order to characterize the accuracy of these devices. These sniffers capture controlled traffic between an access point and a client in individual trace files. By analyzing similarities and differences in these trace files we find that, even in carefully deployed scenarios that include an anechoic chamber, sniffers fail to catch frames that were captured by a neighboring sniffer. We characterize the extent of this loss for different sniffers and analyze loss correlation within a single sniffer and among sniffers.

I. Introduction

In recent years, a number of efforts have performed measurements studies in 802.11 b/g networks with a variety of goals, ranging from wireless channel characterization in production wireless networks [1] to link interference measurement [2], to assessing the degree of standards conformance of 802.11 interface cards [3]. While existing work has shown that coarse-scale sniffer placement [1] and data rate [4] can be crucial factors in the ability to sniff wireless traffic, an implicit assumption has been that sniffers themselves do not introduce significant measurement error.

In this paper, we experimentally investigate the accuracy of three different passive wireless packet capture devices in two controlled, interference-free scenarios -an office scenario with negligible observed traffic other than the flows being measured, and in an anechoic chamber - in which three collocated sniffers capture frames being sent between an access point and a receiver. We characterize the extent to which a sniffer fails to capture ("misses") a transmitted frame that is captured at another sniffer and analyze the correlation among missed frame events at the sniffers. We also investigate the variability of the missed frame rate within each receiver, the extent to which this rate is location-dependent, and the effect of frame transmission rate and frame type (RTS/CTS, beacon, DATA, ACK) on frame miss events at the sniffers. We find that there can be considerable variation in even the average rate of missed frames among the capture devices, and show (via hypothesis testing) that missed packet events among receivers are essentially independent. As a consequence, unioning the set of received packets at two sniffers provides an extremely accurate estimate of the number of frames successfully transmitted. Together, these results provide valuable insights into the accuracy of frame capture and loss measurements reported by wireless packet sniffers.

The reminder of this paper is structured as follows. In Section II, we describe the testbed scenarios used in our study. In Section III, we study the capture performance of the individual devices; and in Section IV characterize the accuracy of the combined measurements from multiple devices. Related work in the area of wireless network measurements is presented in Section V. Section VI summarizes this paper and discusses future research.

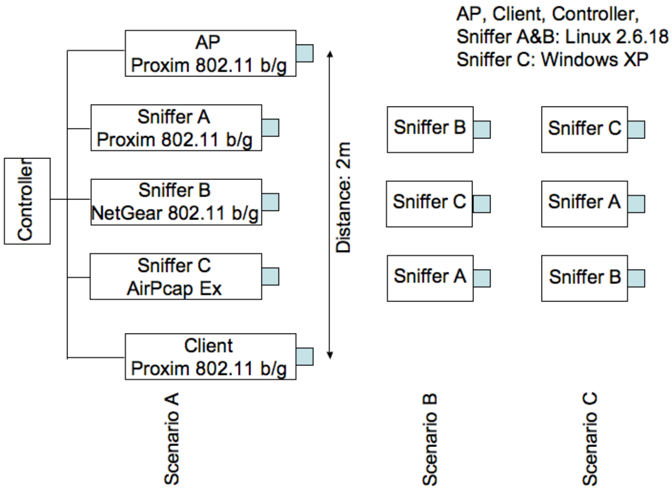

Fig. 1. Measurement setup

II. Testbed Description

Our testbed (as shown in Figure 1) consists of 5 laptops. Two of these machines are used to set up the controlled experiment, one serving as an 802.11 Access Point, the other one as a client. Both of these laptops have Proxim Orinoco 11b/g PC cards as wireless adapters installed. The other three laptops serve as passive wireless sniffers (i.e., they do not send any frames), two running Linux (sniffers A and B) and the third running Windows XP (sniffer C). Both Linux-based sniffers use PCMCIA wireless cards. Sniffer A uses a Proxim Orinoco 11b/g PC card while sniffer B has a Netgear WG511T PC card installed. The Windows-based sniffer makes use of CACE's AirPcap EX which is a USB 2.0 adapter. All 4 Linux-based laptops use madwifi 0.9.2 as driver for the wireless adapter, while for the Windows-based laptop the Airpcap v3.2.1 driver is used.

One additional desktop machine serves as a controller for the entire experimental setup. All machines are connected via wired Ethernet. This allows for automated execution of the experiments and ensures that the control commands sent from the controller machine do not interfere with the wireless measurements. We deployed our testbed in two different scenarios: in an anechoic chamber (Figure 2 where no frame transmissions except those from our machines could be detected, and in an office scenario (Figure 3) where a negligible number of transmissions from different 802.11 sources could be detected.

Unless otherwise specified, we use the following configuration as the *base case* for our analysis: we use 802.11g mode, with the RTS/CTS mechanism activated, and we set the AP to send UDP packets of 72 bytes at the maximum achievable rate. We make use of the Iperf[1] bandwidth measurement tool to send packets between the AP and the client. The packet capturing process is performed with tshark[2] on the three sniffers, capturing the first 100 bytes of each frame, including all frame headers.

Figure 1 also shows the relative physical placement of the sniffers in the measurement setup. In Section III-A, we investigate the effect of sniffer placement on monitoring performance. We considered 3 different placement scenarios, which are depicted as scenarios A, B, and C in Figure 1. In all three scenarios we use the same set of physical locations, on changing the laptop placements in these locations.

A. Interference measurements

Prior to running our measurements, we measured possible sources of interference. We proceed as follows

[1] http://dast.nlanr.net/Projects/Iperf
[2] http://www.wireshark.org/docs/man-pages/tshark.html

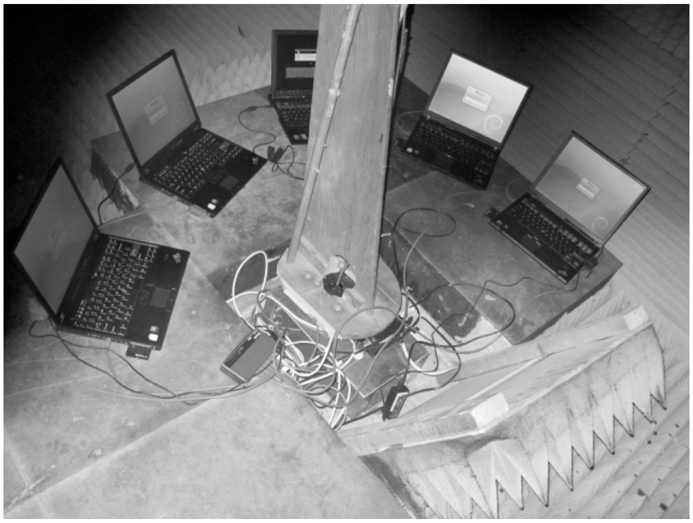

Fig. 2. Anechoic scenario

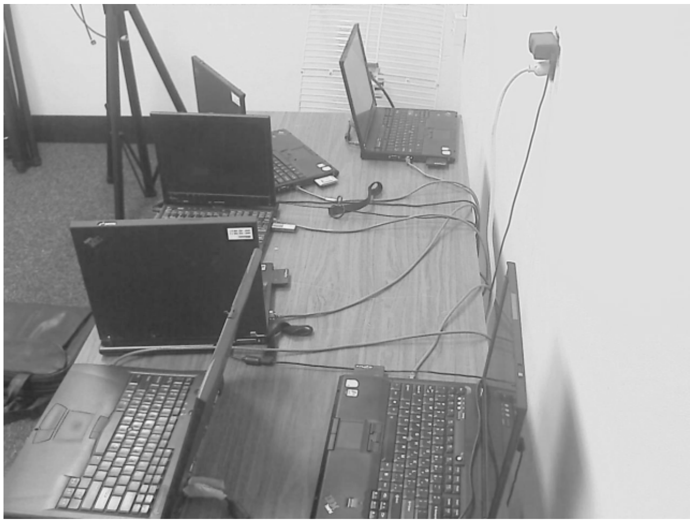

Fig. 3. Office scenario

- We first run a *passive scanning*: we configure a sniffer to listen to any transmission on a channel for 2 minutes, and count the total number of frames captured. We run this scan for all 11 frames, and repeated the process 5 times.
- We the run an *active scanning*, by means of two wireless devices: one configured as an AP, sending beacon frames on a given channel, the other configured as a sniffer. We proceed similarly as before, counting all frames but those beacons sent by our AP. This way we could detect stations that remain *hidden* from passive measurements, and only send data (typically, probe requests messages) in presence of other sources of traffic.

It can be seen from Figures 4 and 5 that interference inside the anechoic chamber is completely negligible, while at the office scenario although measurements are ran during nighttime, some care must be put when choosing the channel to use. Also, it is interesting to note that for some channels there is a non-negligible number of passive nodes that might transmit in presence of an AP.

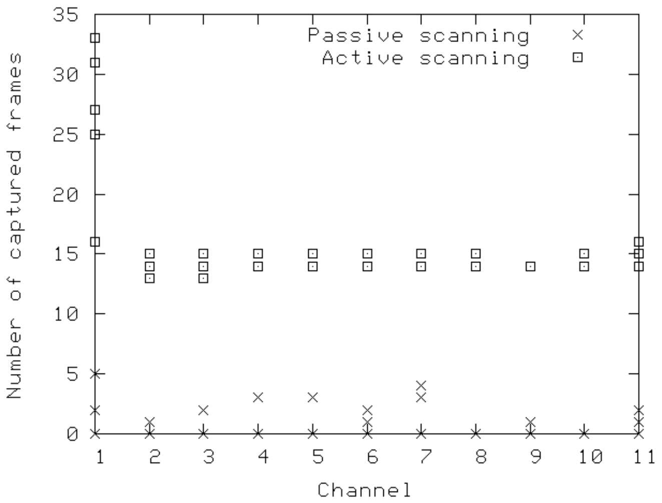

Fig. 4. Measured interference inside the anechoic chamber

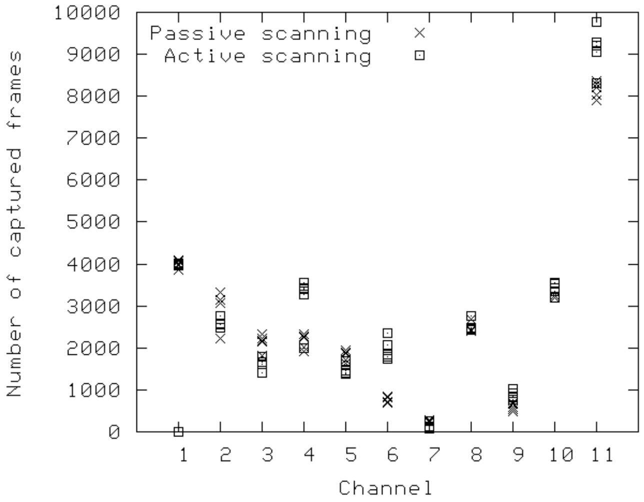

Fig. 5. Measured interference at the office scenario

III. Analyzing the performance of individual sniffers

We begin our study by the common assumption that the sole reason for sniffers to miss frames is a poor radio reception. To analyze to which extent this assumption is valid for each sniffer, we run controlled tests in which we changed one or more parameters of the experiment, e.g. sniffer placement, sending rate or MAC configuration, and assess the accuracy of the sniffer in terms of its *loss rate*. For most of our analyses we will use as performance metric the *estimated data loss rate* (we will later consider non-data frames). Given a sniffer, this loss rate is defined as the ratio between the number of data frames missed by a sniffer over the total number of data frames successfully received by the client. To estimate this loss rate we use the following expression

$$Loss\ rate = 1 - \frac{\#\ unique\ IP\ id's\ captured}{Iperf\ packets\ reported} \qquad (1)$$

Since we are running our controlled experiments in an interference-free scenario with only one active source at a time, even in the event that a frame was not received by the client and needed to be

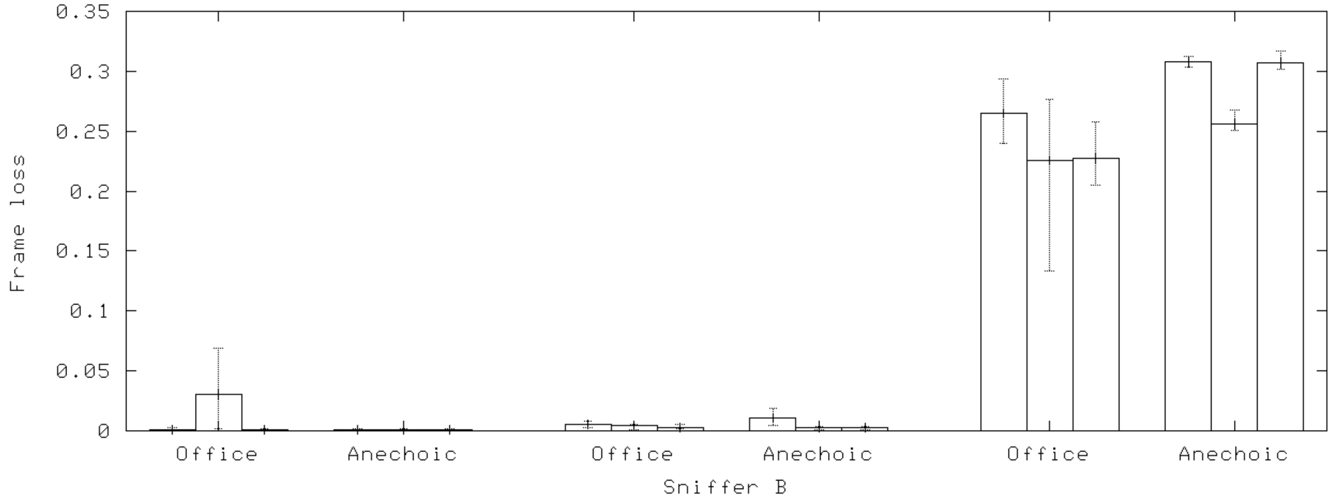

Fig. 6. Effect of scenario deployment on performance. For each sniffer (A,B,C) and environment ('Office' and 'Anechoic') the results for three different deployments are shown.

retransmitted, we observed that the number of retransmissions for the same IP packet never reached the retry limit. Thus, each IP packet was eventually received correctly at the client and thus the number of unique IP packets successfully sent was the same as the iperf-reported number of received UDP packets. Actually we found it surprising that, although we use a similar configuration for both WLAN sender/receivers and sniffers, we only detect losses at the sniffers.

A. Scenario Deployment

We first analyze the effect of sniffer deployment on the monitoring performance. One would expect that, although different sniffers could exhibit different behaviors, a given sniffer should perform consistently long as it is placed close enough to the wireless communication devices (AP and Client in Figure 1). That is, we would not expect small changes in position would have much of an impact on a device's capturing ability. However, we find that this is not the case: performance can vary significantly depending on subtle variations on location (changes on the order of 50 cm).

To quantify the impact of sniffer placement on performance, we proceed as follows. We define three different physical deployments (Scenario A, B and C in Figure 1) for the two environments we used for our test ('Office' and 'Anechoic'). For each of these six possible scenarios we run a series of 50 measurements, where we send approximately 50k data frames and count the number of missing frames at every sniffer. We then compute the average value of this loss rate, and its 0.1 and 0.9 percentile, out of the 50 measurements.

We show results, grouped by sniffer, in Figure 6, using bars for the average loss rate and vertical lines for the percentile values. It is clear from Figure 6 that sniffer C provides the worst performance of all the three regardless of the deployment. Sniffer C's performance also varies depending on the physical location, as the average sniffing ability is better in the Office scenario, but the variability of the results is smaller inside the Anechoic chamber. For the case of the Linux-based devices, sniffer A outperforms sniffer B except for one Office environment case. For this particular scenario, there is a large increase in both the loss rate and variability for sniffer A.

We conclude that the physical deployment of sniffers in a wireless scenario has an unpredictable and non-negligible impact on performance, both in terms of average values and their variability. We have seen that some deployments in the Office environment, for instance, can outperform carefully deployed scenarios in the interference-free environment (the anechoic chamber). Because of this inherent inability

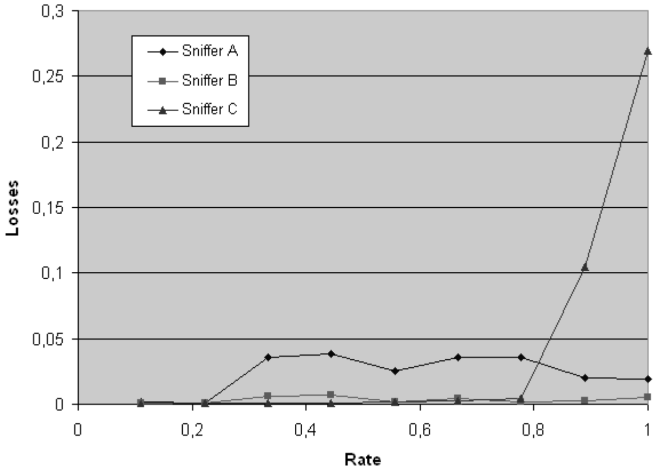

Fig. 7. Effect of traffic rate on performance

of sniffers to capture all data frames that were successfully received at the client, we conclude that one possible way to address this less-than-ideal performance is to sniff using more than one device –an approach we study in Section IV.

B. Assessing the Impact of the Characteristics of the Wireless Communication Under Study

We next consider the extent to which the parameters of the wireless communication under study have an impact on the accuracy of the sniffer. That is, we are interested to assess if, for given physical deployment, the performance of wireless sniffers depends on the characteristics of the wireless communication being monitored.

We first focus on the traffic rate and analyze the relationship between the frame transmission rate and the loss rate. We proceed as follows: for the office environment, we vary the sending rate from 10% of the maximum rate to 100% in 8 steps, capturing approximately 50k data frames for every experiment. We repeat each experiment 10 times, and then plot the average value of the loss rate vs. the relative sending rate in Figure 7.

Figure 7 shows that the performance of Linux-based sniffers remains approximately the same regardless of the traffic rate. For the Windows-based sniffer, as long as the sending rate is not very large its performance is also quite similar to that of Linux devices. However, there is a certain threshold that, once crossed, causes sniffer C performance to drop abruptly (according to the figure, this threshold occurs at approximately 90% the maximum sending rate).

The above analysis was done using our base case (802.11g, RTS/CTS, 100 bytes frames). We next analyze whether changing this configuration results in a change of sniffer performance. With this goal in mind we varied the parameters of the 802.11 communication for a total number of 8 different cases, and run 50 measurements for each of them (in all cases the sending rate was set to the maximum achievable value). The results for the average of these tests are summarized in Table I.

Let us first consider configuration # 7 of Table I. In this case, we use 802.11g mode to transmit 100 byte frames at the maximum achievable rate, using the RTS / CTS exchange before the actual transmission takes place. As a result, the total frame rate (RTS + CTS + Data + ACK) is quite large, and we have the same behavior as in high sending rates of Figure 7. Let us consider now configuration # 8, where only the packet size is changed from 100 to 1500 bytes. For this case, because it takes more time to transmit a data frame, the total frame rate is reduced, therefore leading to improved performance of the Windows device.

TABLE I

LOSS RATE FOR DIFFERENT CONFIGURATIONS OF THE COMMUNICATION.

	802.11b		Sniffer		
#	Mode	IP packet size	A	B	C
1	Basic	100 B	0.00063	0.00069	0.00556
2		1500 B	0.00055	0.00047	0.00293
3	RTS/CTS	100 B	0.00017	0.00016	0.00178
4		1500 B	0.00025	0.00127	0.00059
	802.11g		Sniffer		
#	Mode	IP packet size	A	B	C
5	Basic	100 B	0.00178	0.00076	0.01589
6		1500 B	0.00064	0.00251	0.03922
7	RTS/CTS	100 B	0.00094	0.00293	0.22742
8		1500 B	0.00052	0.00541	0.01703

If we compare sniffer performance for every considered configuration using the 802.11b mode (# 1–4) with the same configuration for the 802.11g mode (# 5–8), we see that results are typically better for the former case. This might be explained because 802.11b mode is not as demanding in terms of sending rates as the 802.11g mode. In any case, it is clear that performance is, in general terms, quite unpredictable. For example, sniffer A outperforms sniffer B for the 802.11b, RTS 1500-byte case (# 4). However, sniffer B outperforms sniffer A for the 802.11g, non-RTS 100-byte case (# 5).

C. *Frame Type Losses*

In 802.11 WLANs, different types of frames can be sent not only with a different modulation scheme (e.g., the modulation rate of a data frame is typically higher than the modulation rate of a beacon frame), but also after a different sequence of events (e.g., ACK frames are guaranteed by the MAC operation to be collision-free in most circumstances, while RTS frames are used to gain access to the medium and therefore can collide). This motivates us to analyze if all frames are equally likely to be lost, or if the loss rate might depend on the frame type.

To analyze the impact of the frame type on the loss rate, we use the data from one of the scenarios in the anechoic chamber as follows. For all measurements for that particular scenario, we merge the tracefiles from each sniffer (similarly to the Jigsaw scheme[1]). For this merging to work, first we need to asses the accuracy of timestamping.

D. *Timestamps*

- We first measured the cumulative distribution function (CDF) of the time between a data frame and the corresponding acknowledgement frame. According to the 802.11 standard, this time is fixed, given by $SIFS + T_{PCLP} + ACK/R_{ack}$. We plot the results for these values (for both mode 802.11b –left– and 802.11g –right–) in Figure 8. It can be seen from the figure that, for all interfaces, measurements are both accurate and precise.
- To look further into these measurements, we measure also the time between one acknowledgement frame and the next data frame, given no beacon nor retransmission took place. According to 802.11 standard, this time should be given by $DIFS + U(0, CW_{min})T_{slot} + Ts$. The CDFs for this measurement are plot in Figure 9. Not only could we distinguish the maximum number of slots for each modulation scheme (16 for 802.11g and 32 for 802.11b), but also its size (9 μs and 20 μs, respectively), without the need of a proprietary hardware as in [3].
- However, we faced some problems with one of the adapters (Sniffer C). As it can be seen in Figure 10, Sniffer C timestamps –using the driver provided with the hardware ("old driver" in the Figure)–

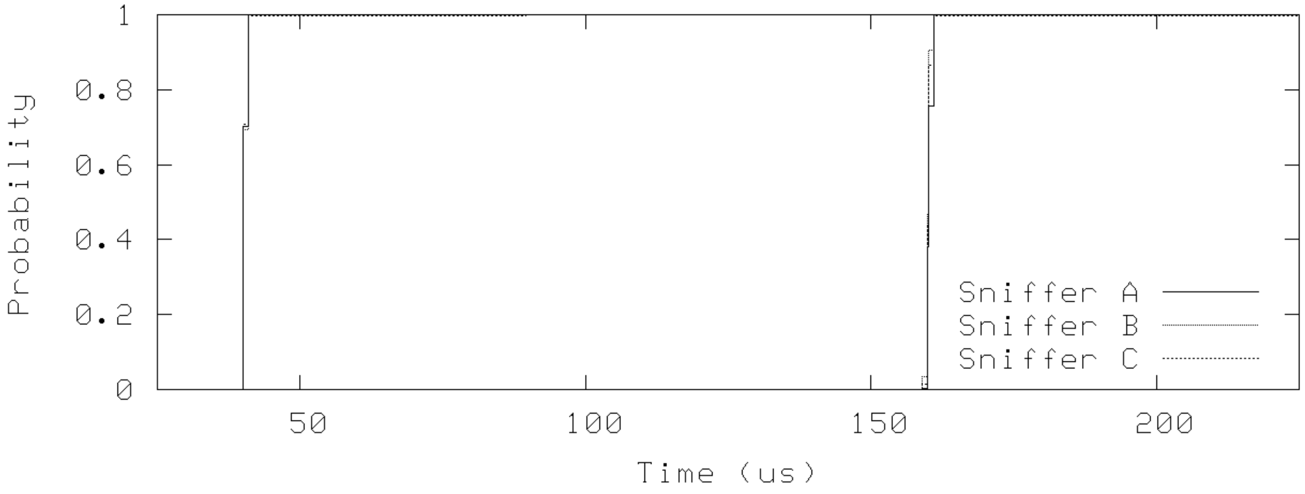

Fig. 8. Time between a data frame and the corresponding ack frame

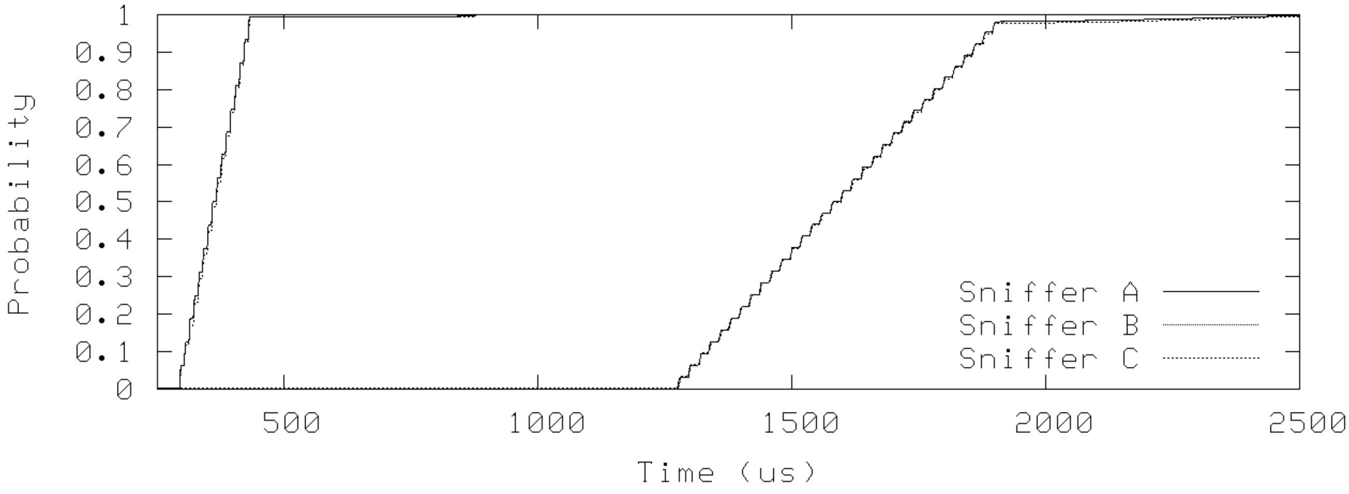

Fig. 9. Time between an ack frame and the next data frame

are not strictly increasing (as one could expect), but rather some kind of periodical offset is added and subtracted about every 30 ms. The vendor provided us with a new version of the driver ("new driver" in Figure 10) that did not show this performance. However, due to time constrains we could not use the tracefile from Sniffer C in the merging procedure, as we rely heavily on accurate delays and the "old driver" provided both negative and huge delays.

E. Merging algorithm

The merging algorithm basically works as follows

1) Search for a common uniquely identifiable frame, i.e., a non-retransmitted IP packet or a Beacon frame, present on all three tracefiles. Use this frame as the first frame of the merged tracefile, and to set a time reference for relative delays.
2) Look for the *next* frame on the tracefiles, and pick the one with smallest relative delay.
3) Check the next frame. In case it is from the other two tracefiles, make sure that it is not the same frame (within a small time range), and mark it as missing.

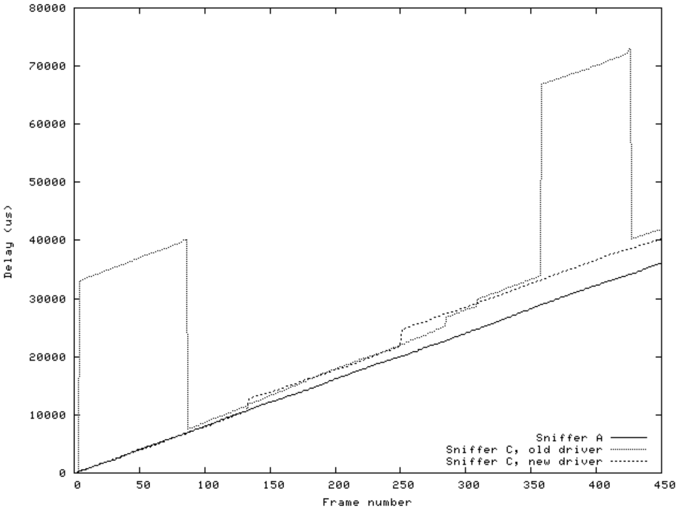

Fig. 10. Timestamps performance for Sniffer A and Sniffer C, old and new driver

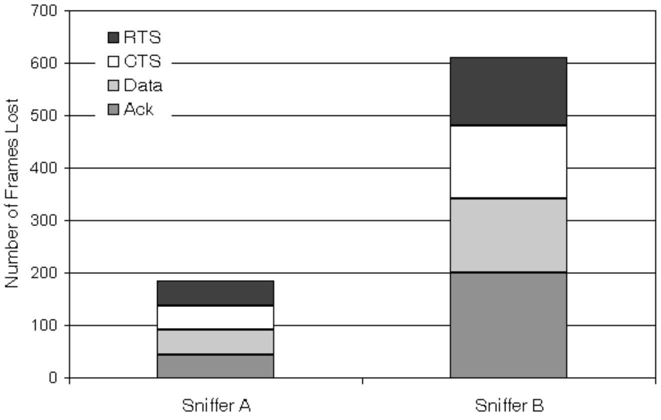

Fig. 11. Losses for different frame types

4) In case the same frame is captured by the three tracefiles, set this new as time reference.

F. Results

We plot in Figure 11 the average number of frames lost, by frame type, for the two Linux sniffers. Note that our measurements are performed in a WLAN with a single sending station in an interference-free environment (i.e., few radio losses and no collisions), therefore the number of RTS, CTS, Data and ACK frames is roughly the same (this is not the case for Beacon frames, but we found these were rarely lost).

It is clear from Figure 11 that sniffer A significantly outperforms sniffer B, as the total number of frames lost is three times smaller. But there is also a difference in terms of which type of frame is missed: while at sniffer A all frame types are equally likely to be lost, at sniffer B ACK frames have a higher chance of being lost than any other frame. Note that both CTS and ACK frames are sent in similar manner (with a lower modulation rate and immediately following a previous frame reception, with guaranteed medium availability).

Therefore we conclude that, for some sniffers, there might be a non negligible dependence on the frame type in the loss process. This is a quite unexpected result that could introduce a non significant bias on the experimental evaluation of WLAN performance. For example, if the presence of an ACK is taken

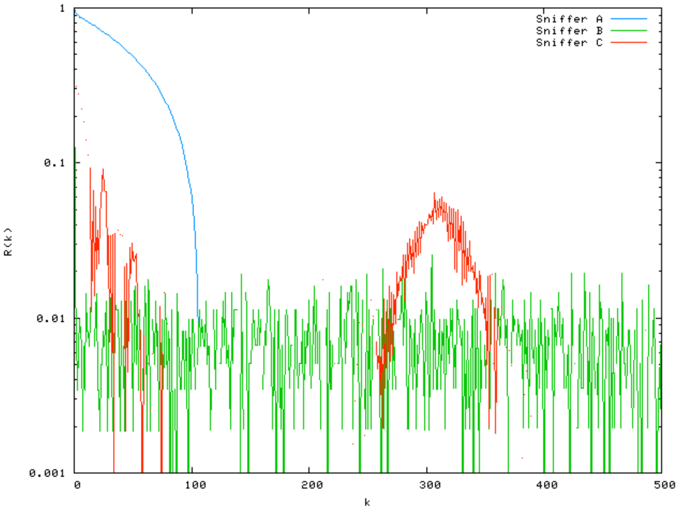

Fig. 12. Autocorrelation of loss processes

as an indication of a successful communication [5], this uneven loss process between frame types could introduce significant bias in the measurement process, particularly in scenarios where most frames are successfully received.

G. Autocorrelation of Loss Process

We next focus on one run of the series of measurements made in the anechoic chamber, and analyze the autocorrelation of the loss process. We first build the list of the n unique IP identifiers that were sent with the controlled experiment, and based on this list we build for each sniffer a sequence x of events, where '1' represents a sniffer miss, and '0' represents the sniffer captured that frame. We then compute the following estimator of the autocorrelation

$$R(k) = \frac{1}{(n-k)\sigma^2} \sum_{t=1}^{n-k} (x[t] - \mu)(x[t+k] - \mu) \tag{2}$$

where μ and σ^2 are sample estimators for the mean and standard deviation of x.

In Figure 12, we plot $R(k)$ for the three sniffers and different values of k (we obtained similar results when we used different sets of data). The results can be summarized as follows:

- Most of the losses of sniffer A (about 100) happen in a burst, as its autocorrelation slowly decreases from $R(0) = 1$ to $R(100) = 0$.
- No pattern can be seen in the autocorrelation for sniffer B.
- Sniffer C, the Windows device, has a periodical loss process with a period of approximately $k = 300$ data frames.

We conclude that the autocorrelation of the loss process also varies among sniffers, so one should take care when using two identical devices to improve the capture ability, as it could be the case that both sniffers tend to miss the same frames. We address these questions about loss correlation between sniffers and benefits of merging the traces from multiple packeteapture devices in the next section.

IV. Combined Measurement Analysis

The main result from the previous section can be summarized as follows: even for carefully deployed scenarios, a single sniffer is not enough to capture all frames that were sent in a wireless communication. We have seen that any COTS sniffer can have an unpredictable loss rate that depends both on the physical

TABLE II

Cross-Correlation Analysis

Sniffer pair	r	$Z(-0.3)$	$Z(0.3)$
(A, B)	−0.059	1.72	−2.52
(A, C)	−0.071	1.64	−2.83
(B, C)	−0.103	1.42	−2.61

location of the sniffer and the characteristics of the wireless communication under study. Furthermore, even the autocorrelation of the loss process can be different depending on the device. Therefore, one way to improve the capturing ability is to use more than one sniffer and, after measurements are made, merge the collected tracefiles.

However, for the merging approach to work, losses should not be highly correlated across sniffers. Otherwise, if two different sniffers tend to miss the same frames, little if any gain would be realized. Therefore, we first analyze the correlation between losses from different sniffers, and then evaluate the gain obtained from merging.

A. Loss Correlation Analysis

We focus on a particular scenario inside the anechoic chamber, and analyze one series of 50 runs of experiments as follows. Each of this 50 experiments provides a 3-tuple with the average loss rate for each sniffer ($loss_A$, $loss_B$, $loss_C$). In Figure 13 we plot the $loss_C$ values vs. the $loss_B$ values. We also plot in Figure 14 the $loss_B$ values vs. the $loss_A$ values. Little, if any, correlation is apparent from the figure (similar figures are obtained for different scenarios and pairs of sniffers, but due to lack of space we do not show them).

We next compute the (Pearson product-moment) correlation coefficient r, defined for two sequences x and y of variables of length n as

$$r_{xy} = \frac{\sum x_i y_i - n\bar{x}\bar{y}}{(n-1)s_x s_y} \tag{3}$$

where $\bar{x}$ ($\bar{y}$) and s_x (s_y) are sample estimators for the mean and standard deviation of x (y). We show the results of r for the three possible pairs on the first column of Table II. As absolute values for correlation are *small*, we could argue that variables are not strongly correlated. However, to have some statistical confidence in this statement, we apply Fisher's r to z transform [6], defined as

$$z = \frac{1}{2}\log\frac{1+r}{1-r}, \sigma_z = \frac{1}{\sqrt{N-3}} \tag{4}$$

where N is the number of samples. With the aid of Fisher's transform, we can run hypothesis tests on the correlation coefficient by means of the *standard score* (or Z *score*). This way, we can test if r is significantly different from $|\rho| = .3$[3], with the results shown on the third and fourth column of Table II. As all results fall outside the critical values of -1.28 and 1.28, we can reject with 90% confidence the hypothesis that correlation is larger than $|\rho| = .3$ and therefore conclude loss rates are weakly or not correlated.

B. Collective Sniffing

In Section IV-A, we have seen that losses are essentially independent among the three sniffers under study. Therefore, there will always be a gain in performance when merging the tracefiles from any two

[3] This is a somehow arbitrary threshold to distinguish between *small* and *medium* correlated variables, proposed in Cohen, J. (1988), Statistical power analysis for the behavioral sciences (2nd ed.) Hillsdale, NJ: Lawrence Erlbaum Associates.

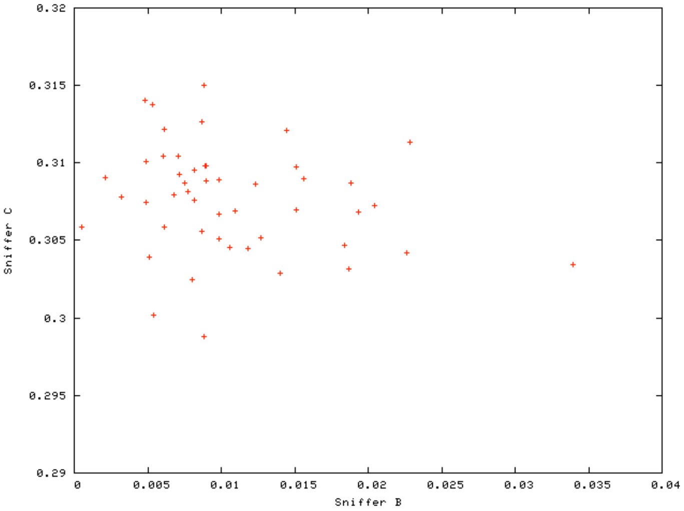

Fig. 13. Correlation plot of loss rates

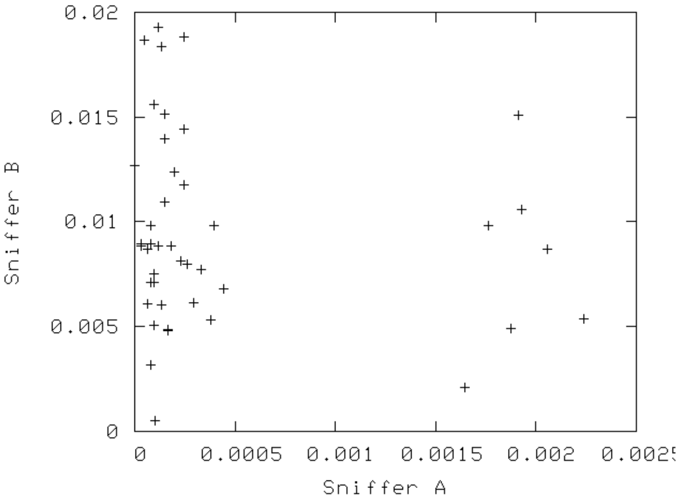

Fig. 14. Correlation plot of loss rates

sniffers. Furthermore, because losses are independent, the probability of missing a frame in the merged file is given by the product of the loss probability from each of the sniffers involved.

To validate the above we consider one series of experiments from the anechoic chamber (the same one used on the previous section). We compute the average loss rate ('measured losses'), first for each sniffer on its own and then for the three possible pairs of merged tracefiles (note that, as we are counting unique IP identifiers, there is no need to actually merge the tracefiles, so we can use sniffer C). We then compare these results with the expected values assuming independent losses. That is, we compute the probability of a frame loss for a merged scenario $p(A+B)$, and compare it with the probability that would be obtained in case losses were independent, $p(A)p(B)$, where each of these probabilities is also obtained from measurements.

We present our results in Table III, where the average loss rate for the 50 measurements is shown in the second column. The third column shows, for the cases of two merged tracefiles, the "expected" values we would obtain by multiplying the loss rates from the single sniffer scenarios. For example, for the last row of the column, $0.00975(B+C) = 0.03174(B) \times 0.30723(C)$. We see that these two values are quite

TABLE III

PERFORMANCE OF DIFFERENT SCHEMA

Single sniffer	Measured losses	
A	0.00226	
B	0.03174	
C	0.30723	
Two sniffers	Measured losses	Expected
A+B	0.00006	0.00007
A+C	0.00063	0.00069
B+C	0.00969	0.00975

similar for the three possible pairs, as expected given our analysis of the previous section.

Therefore, as long as the performance of sniffers is reasonably good, only a few capturing devices are needed to get a very accurate merged tracefile for a given scenario.

V. RELATED WORK

There has been some previous work concerned with obtaining the most accurate picture of the behavior of a Wireless LAN. Yeo et al. [7], [8], [9] were one of the first to report experiences and pitfalls from sniffing deployments, and to advocate for the merging of tracefiles from sniffers placed far apart to improve capture ability. However, because data is sent from both the AP and the client, the relative position of sniffers and the direction of the transmission are very related (probably because of the so-called *capture effect*). Our work uses a more careful deployment, with results taken from measurements inside an anechoic chamber, to analyze differences in performance between sniffers placed closed together, providing not only average values of performance but also insightful results about the nature of the loss processes and their relation.

Jigsaw [1] is a distributed wireless monitoring infrastructure that is deployed in an Office scenario to aid in networking diagnosis. Besides the fact that we also merge data from different sniffers, our goal is to investigate the limits of the measurement process (rather than to perform network diagnosis) using single and multiple devices. In [4], the authors have analyzed single and multiple sniffer performance under extremely high frame sending rate scenarios, although their focus was on the hardware architecture limits rather than on the measurement process itself.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we have analyzed the ability of sniffers to accurately capture frames sent in a wireless communication. We have observed that there is always an unpredictable loss at packet sniffers of packets that are received at the monitored receiver and other nearby sniffers, which can depend on the various characteristics of the scenario (position, rate, modulation scheme, etc). These unknown performance limits bounds the accuracy of measurements derived from analysis of experimental tracefiles, as the tools used to capture traffic introduces unpredictable losses. As future work we plan to analyze the extent to which these losses impact experimental measurements. We also plan to work on mechanisms to lessen these effects. More specifically, we are developing a timestamp-based interference algorithm to improve single-sniffer measurements.

ACKNOWLEDGMENT

This work was funded in part by the National Science Foundation under grants, EEC-0313747 001, ANI-0325868, and EIA-0080119, and by the Ministry of Education and Science of Spain, under a José Castillejo grant. We thank Ian Ricci for his support in setting up the testbed infrastructure and Gene Losik and Dan Schaubert for providing access to the anechoic chamber.

References

[1] Y.-C. Cheng, J. Bellardo, P. Benk ö, A. C. Snoeren, G. M. Voelker, and S. Savage, "Jigsaw: solving the puzzle of enterprise 802.11 analysis," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, pp. 39–50, 2006.

[2] J. Padhye, S. Agarwal, V. N. Padmanabhan, L. Qiu, A. Rao, and B. Zill, "Estimation of link interference in static multi-hop wireless networks," in *Proc. IMC 2005*, 2005.

[3] G. Bianchi, A. D. Stefano, C. Giaconia, L. Scalia, G. Terrazzino, and I. Tinnirello, "Experimental assessment of the backoff behavior of commercial ieee 802.11b network cards," in *INFOCOM*. IEEE, 2007, pp. 1181–1189.

[4] M. Portoles, M. Requena, J. Mangues, and M. Cardenete, "Monitoring wireless networks: performance assessment of sniffer architectures," in *Communications, 2006. ICC '06. IEEE International Conference on*, vol. 2, June 2006, pp. 646–651.

[5] D. Giustiniano, D. Malone, D. Leith, and K. Papagiannaki, "Experimental assessment of 802.11 mac layer channel estimators," *Communications Letters, IEEE*, vol. 11, no. 12, pp. 961–963, December 2007.

[6] P. R. Cohen, *Empirical Methods for Artificial Intelligence*. Cambridge, Massachusetts: MIT Press, 1995.

[7] J. Yeo, S. Banerjee, and A. Agrawala, "Measuring traffic on the wireless medium: Experience and pitfalls," 2002. [Online]. Available: citeseer.ist.psu.edu/yeo02measuring.html

[8] J. Yeo, M. Youssef, and A. Agrawala, "A framework for wireless lan monitoring and its applications," in *WiSe '04: 3rd ACM workshop on Wireless security*, 2004.

[9] J. Yeo, M. Youssef, T. Henderson, and A. Agrawala, "An accurate technique for measuring the wireless side of wireless networks," in *WiTMeMo '05: Workshop on Wireless traffic measurements and modeling*, Berkeley, CA, USA, 2005, pp. 13–18.