Measurement-Based Wireless LAN Troubleshooting

Sebastian Felis, Jürgen Quittek and Lars Eggert
NEC Europe Ltd., Network Laboratories, Kurfürstenanlage 36, 69115 Heidelberg, Germany
{felis|quittek|eggert}@netlab.nec.de

***Abstract*— Wireless networks based on the IEEE 802.11 family of standards are becoming increasingly common. Their wide-scale deployment introduces a set of new operational problems concerning connectivity, security and quality-of-service. This paper describes a set of measurement tools for troubleshooting and securing a set of geographically overlapping wireless networks in infrastructure mode and in *ad hoc* mode. These tools improve on existing mechanisms by analyzing link-layer sequence numbers, deducing link congestion based on retransmission traces and by detecting cross-channel interference based on observing sets of neighboring or overlapping channels at the same time. The paper illustrates the effectiveness of these new measurement techniques through a series of short, real-world experiments.**

I. Introduction

WITHIN the last few years, wireless local area networks (WLANs) based on the IEEE 802.11 family of standards [3] have become the most popular technology to provide wireless network connectivity. The initial set of capabilities was simple and geared towards quick installation of small office or home networks, resulting in low equipment prices and quick adoption. Today, almost all new mobile computers come equipped with WLAN interfaces. Many other devices, such as cell phones and handheld computers, are beginning to converge on WLAN as well, bypassing other technologies for providing local wireless connectivity, such as infrared serial links or Bluetooth.

The result of this development is that WLAN networks are becoming ubiquitous. Despite the limited coverage range of each individual access network, multiple, different WLAN networks are oftentimes accessible by a single end system, especially in densely populated areas, such as office or apartment buildings.

This dense deployment causes performance and stability problems for the individual networks due to interference and other reasons. One approach to dealing with these issues is a common management system that automates configuration and management of a set of WLAN networks as an integrated whole. However, these systems assume that a single entity has operational control over the whole set, which is frequently not the case. Furthermore, neighboring, foreign networks or misconfigured local clients cause problems outside the influence of a management system.

Consequently, management of a deployed WLAN network, independent of the number of involved base stations, requires additional measures for efficient operation. Measurement-based troubleshooting of the network is one important component to achieve this goal.

Analysis of traffic measurements of a deployed wireless network can pinpoint several causes of problems. First, inter-channel and cross-channel radio interference can significantly decrease the effective data rate of the network. Second, when multiple wireless networks that belong to different entities overlap, security becomes of utmost importance. This includes verifying that all friendly networks are sufficiently protected (*i.e.*, access is authenticated and/or data is encrypted) and detecting intrusion attempts. These two groups of problems are related: even unsuccessful intrusion attempts may interfere with network operation.

The tools presented in this paper were motivated by the need to manage and troubleshoot NEC's local wireless networks. In addition to a production network that provides wireless connectivity throughout the laboratory, more than ten experimental infrastructure and *ad hoc* networks compete for spectrum. Additionally, non-NEC networks inside and outside the building create additional problems. Managing the operation of this diverse set of networks requires considerable effort.

Consequently, tools were designed that can automatically determine the most common causes of problems based on ongoing network measurements. One such mechanism detects transient congestion or interference by observing frame retransmissions. Another analyzes the link-layer sequence numbers of traffic traces and can, for example, detect spoofed frames more reliably. Additionally, these tools can aid a human operator during complicated investigations by providing metrics that are more meaningful than raw traffic traces. One such metric is sequence number plots, which illustrate transmission rates over time and can help identify faulty hardware.

Section II discusses the detailed requirements of such measurement tools. Section III describes the proposed solution and discusses the capabilities of the current prototype implementation. Section IV illustrates the effectiveness of the measurement tools through a number of real-world experiments. Finally, Section V gives an overview of planned future work and concludes this paper.

II. Measurement System Characteristics

This section discusses different characteristics for a measurement system for automated and semi-automated network troubleshooting. Section III then describes the capabilities of the designed system and the current prototype implementation.

Measurement systems take many physical forms, ranging from small, mobile handhelds with limited compute power to powerful stationary measurement stations. The basic tradeoff is one between compute power, limiting the traffic volume that can be captured and analyzed, and mobility, limiting the ability to capture traffic many points within the coverage area. Hybrid

systems are possible, where mobile sensors gather traffic traces for analysis at more powerful, stationary devices.

Today, available off-the-shelf mobile systems typically cannot capture traffic on many different channels at the same time, due to compute and data bus limitations, but also size considerations. More powerful stationary workstations, on the other hand, feature faster processors and buses that can support many parallel WLAN interface cards to capture traffic on many different channels in parallel. An intermediate approach is to switch a single interface card between channels when a realtime analysis indicates that cross-channel interference may be the cause of detected problems. However, switching channels incurs a delay, which makes this technique not suitable for very brief events.

A second consideration is the types of measurements – active or passive – a station performs. Active measurement techniques inject traffic into the network to observe certain behaviors. Passive measurement techniques never inject traffic of their own and instead monitor any already present traffic. The set of detectable events is thus potentially smaller for the latter, because the present traffic patterns may not exhibit all existing issues. On the other hand, active measurement techniques are themselves observable by others on the network.

Many network monitoring tools, such as *tcpdump* [12], focus on network- and higher-layer information and do not provide access to all link-layer information contained in a traffic trace. However, the link-layer headers oftentimes contain essential information that can aid in identifying specific issues. It is therefore important that the capture tool record and provide link-layer information to the analysis process. For example, 802.11 data and management frames contain *retransmission indicators* that indicate whether the current frame is a retransmission of an earlier one. Frequent retransmissions can indicate transient connectivity problems, as described in Section IV.

As mentioned in Section I, tools that can automatically determine common problems and/or can aid human operators in their troubleshooting can significantly reduce maintenance efforts. For example, metrics that abstract from he underlying raw data can help pinpoint exact causes of problems.

Similarly, support for a notification system to alert human operators to the presence of abnormalities in the network is a useful feature of a measurement and analysis system. An extended notification system for other events, such as appearance and disappearance of end systems, channel or data rate changes or changes to the security parameters, is another useful utility function. Other utility features include persistent storage of at least selected traffic samples for later autopsy or integration with advanced intrusion detection systems and honey pots, such as *snort* [1] or *honeyd* [2].

III. Solution

At the core of a WLAN measurement system is capture of an uninterrupted stream of network traffic. Network interfaces that provide connectivity to an end system cannot continuous monitoring. Consequently, dedicated capture interfaces that served as WLAN monitoring probes are required.

Many different WLAN tools could use these dedicated network interfaces for a variety of active and passive measurements and analyses, including traffic tracers such as *Kismet* [14] or *NetStumbler* [15] and password crackers such that implement known security attacks [10], such as *AirSnort* [16] or *WEPcrack* [13]. Commercial solutions for analysis, management and intrusion detection are *AirMangnet* [17] or *AirDefense* [18].

Kismet, for example, is a wireless network detector, sniffer and intrusion detection system. Based on a client/server architecture, simple measurement sensors relay captured traffic to a central server. *Kismet* classifies networks by SSID and can plot the geographic location of stations via GPS. It supports sniffing with multiple WLAN cards and can extract device-specific information such as signal-to-noise ratios. It implements a flexible notification framework that alerts based traffic patterns and a multitude of intrusion events.

Although some of these features are useful for the desired measurement system, *Kismet* does not offer other required capabilities. It does not store physical-layer information with a traffic trace, does not offer a user-friendly interface to analyze the measured data, does not support MAC-address-based notifications or retransmission and sequence number analysis.

Consequently, the measurement system presented in this paper is not based on *Kismet* but on the command line version of *Ethereal*, called *tethereal* [11]. Tethereal captures raw traffic frames from a dedicated measurement interface, including physical-layer information provided by the device driver, such as signal and noise levels. An *RRDtool* [19] database stores the data as time series for analysis by a set of custom Perl scripts.

The portable measurement hardware can only support a single measurement interface due to processing and bus limitations. Consequently, the measurement interface cycles through all channels with a switch time of 0.2 seconds. Channel switching incurs a delay during which no frames are captured. (As mentioned above, a future, more powerful measurement station will use multiple measurement interfaces to monitor sets of related channels concurrently.) To prevent livelock and maintain timeliness of the system in high-bandwidth scenarios, the measurement system includes a capture shaper that drops frames when the analysis lags too far behind the live capture stream.

For each station that is detected, the measurement system collects additional data such as mode (infrastructure or *ad hoc*), frame and byte counts, associated access point information such as the Service Set Identifier (SSID), physical-layer information such as signal strength and statistical information about higher-layer protocol use.

In addition to many other measurement systems, this system also collects sequence number information for each captured frame. Analysis of patterns in the 12-bit 802.11 sequence numbers of captured frames is an effective technique to determine the presence of various anomalies and attacks [6] [7], as the results in Section IV illustrate.

Additionally, the measurement system deduces link-layer retransmissions based on observing repeated transmissions

of *retry* frames with identical sequence numbers. Frequent retransmissions may indicate connectivity issues. In order to analyze the causes of connectivity issues more thoroughly, the system monitors network activity in interfering channels when retransmission activity increases past a configurable threshold. Because the current system can only support a single measurement interface, it switches through the appropriate related channels rapidly to sample their activity.

As a basic intrusion detection mechanism, the system maintains a database of known MAC addresses that it correlates with the IEEE's "organizationally unique identifier" list [5]. This allows it to identify new MAC addresses and determine whether they are likely spoofed. If the rate of occurrence of new MAC addresses is past a configurable threshold, the system notifies an administrator.

Finally, the measurement system is accessibly through a web-based user interface. Figure 1 shows one example page of this interface. The interface provides point-and-click visualizations of live measurement results, including traffic graphs by network, channel or node, detailed node information including manufacturer, uptime, associated access point, frame counts, signal-to-noise ratios, mean throughputs and higher-layer protocols information.

Note that the main focus of the tool presented in this paper is on WLAN measurements, analyses and visualizations to aid connectivity troubleshooting. Although intrusion detection is relevant in troubleshooting network problems and the tool includes a mechanism based on MAC prefixes and sequence numbers [6], it does not include additional security mechanisms such as [8] [9].

IV. Experimental Analysis

This section illustrates the operation of the measurement system described in Section III through three analysis of real-world data gathered at the NEC Network Laboratories. In all cases, the advanced metrics and visualization capabilities of the measurement tool enabled administrators to determine causes for previously unknown events in the network.

The first example in Section IV-A illustrates how the measurement system uses sequence number analysis to detect that an otherwise well-behaving node is also injecting spoofed attack traffic into the network.

Section IV-B presents a second example that illustrates periodic interferences in the lab network. Output from the measurement tool made it apparent that WLAN stations built into the trams that periodically stop in front of the NEC office caused these disturbances.

Finally, Section IV-C discusses the detection of connectivity issues based on repeated retransmissions.

A. Intrusion Detection Based on Frame Sequence Numbers

This section discusses a technique for intrusion detection based on the sequence numbers of monitored WLAN frames. Because of their predictable and hard to spoof sequence, frame sequence numbers can act as fingerprints that uniquely identify frames sent by a single node over a period of time. This property makes them useful to detect MAC address spoofing.

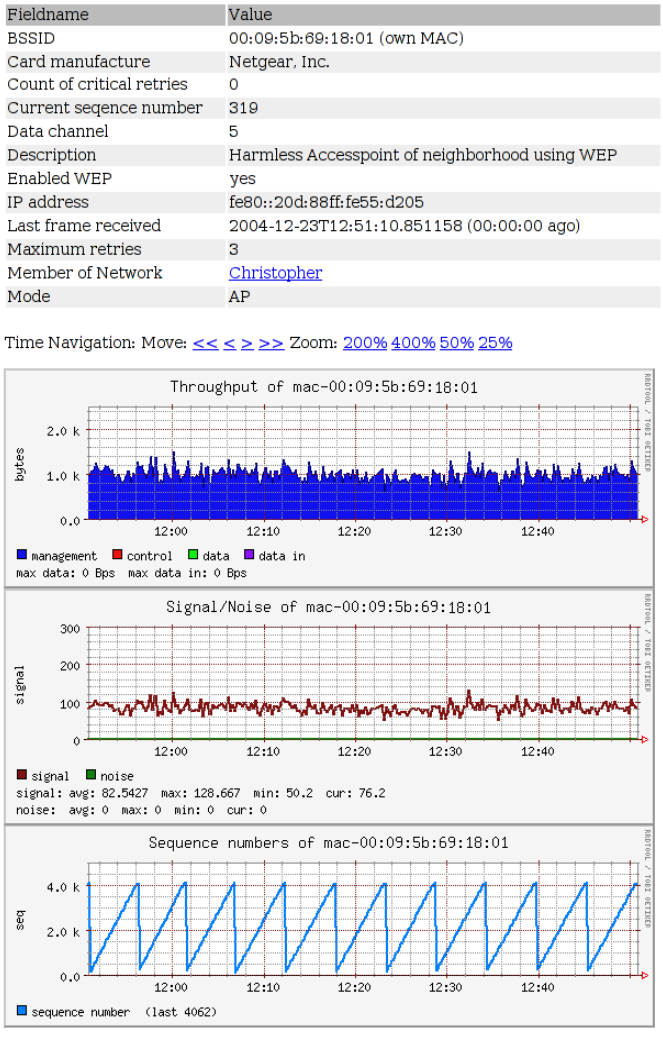

Fig. 1. Web-based user interface of the measurement station showing collected MAC statistics together with throughput, signal strength and sequence number information.

In the 802.11 protocols, a unique sequence number identifies each individual frame sent by a single node to allow detection of duplicates. Sequence numbers are 12-bit counters that monotonically increase from 0 to 4095 and wrap around at overflow. When a network interface starts or is reset, the sequence number counter starts at zero. The minimum time for sequence numbers to wrap around is under one second, but it can be indefinitely longer depending on packet size, send rate and link speed.

According to the 802.11 standard, the sequence number counter should be readable but not writable by software. Consequently, they are difficult to spoof without custom hardware. MAC addresses, on the other hand, can be easily spoofed by software [6]. Even when a station spoofs its MAC address, the sequence numbers of frames sent with a spoofed MAC address will still continue that stations sequence number pattern. This makes frame sequence numbers much stronger identifiers for specific stations than MAC addresses.

Figure 2 shows a plot of frame sequence numbers over time for a single node with MAC address *A*. At the beginning of the

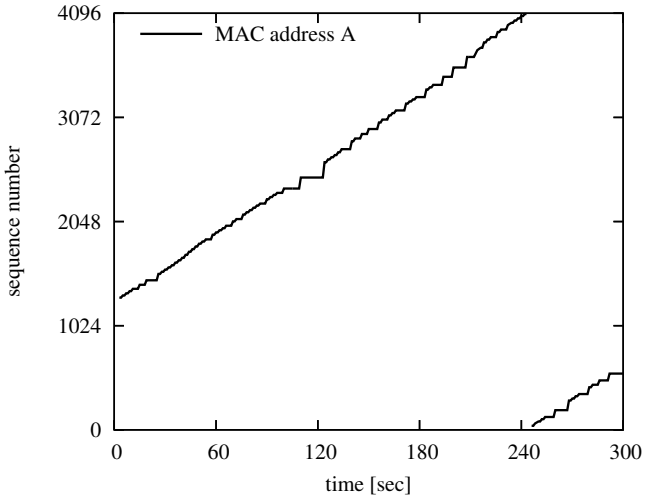

Fig. 2. Plot of sequence numbers over time for a single node with MAC address *A*.

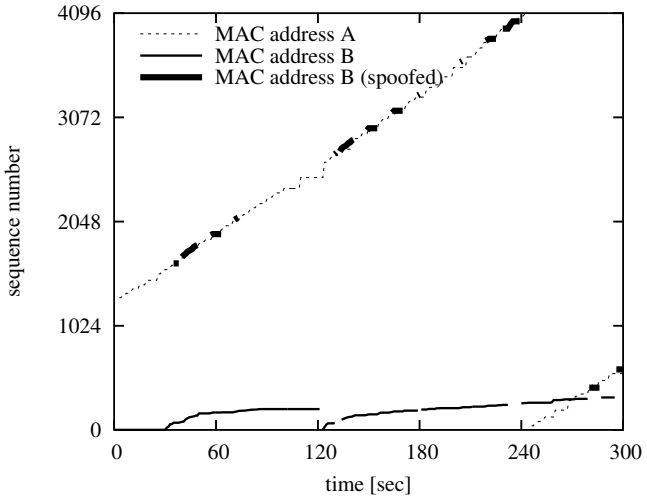

Fig. 3. Plot of sequence numbers over time for two nodes with MAC addresses *A* and *B*.

measurement, the current sequence number is around 1200. Around 110 seconds into the measurement, the node stops sending for around ten seconds (indicated by the horizontal part of the curve) before continuing. At second 243, the sequence numbers wrap around and restart at zero.

Figure 3 shows a similar sequence number plot with traffic from two nodes with MAC addresses *A* and *B*. The sequence number curve for MAC address *A*'s at the top of the graph, the plot for MAC address *B* is mainly at the bottom.

Note that although the two sequence number progressions are clearly distinct, a number of packets are visible that appear to originate at MAC address *B* but have sequence numbers that fit with node *A*'s sequence number pattern. (Figure 3 illustrates these as thicker line segments overlaying node *A*'s curve.)

This trace illustrates how an otherwise well-behaving node with MAC address *A* periodically spoofs traffic to make it appear as if it came from node *B*. Without an analysis based on frame sequence numbers, these spoofed packets are difficult to detect; even more difficult is to determine which station originated the spoofs.

As an aside, a more detailed investigation into the contents of the spoofed packets showed that they are Microsoft NetBIOS packets. The authors are currently investigating the exact nature of this attack traffic.

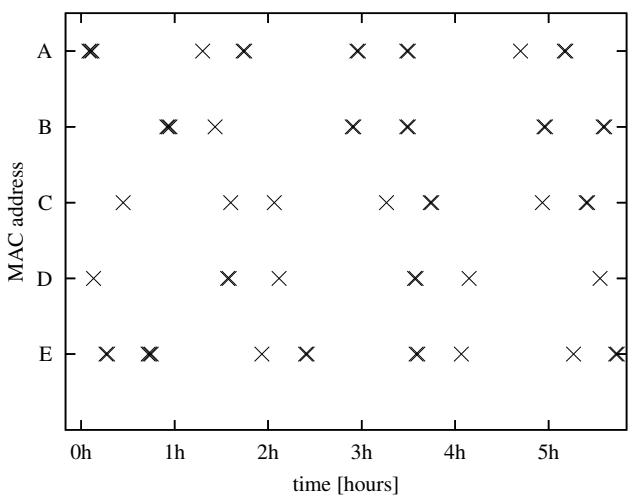

Fig. 4. Periodic probe requests from five different MAC addresses, belonging to five local tram trains that regularly stop in front of NEC's office.

B. Pinpointing Periodic Interference

This section presents a second example that illustrates periodic interferences in the lab network and shows how the measurement system was instrumental in determining their cause.

Figure 4 illustrates periodic probe requests from five different MAC addresses (*A* - *E*) that are recurring periodically over a duration of six hours. These MAC addresses all try to associate with SSID "HSB_WLAN", which is not one of the local networks in the lab. (Note that the slightly bolder crosses in Figure 4 are due to overprinting of multiple, different probe messages.)

The name of the network, however, and the fact that these probes are extremely periodic, allowed administrators to correlate the probes with the arrival of trams at the tram stop in front of NEC's office. [1] Apparently, the trams use WLAN to connect to the operator network at particular stops along their route.

The unique MAC addresses broadcast by the individual trams allow monitoring of their movements. Each tram has a periodic, constant schedule and reverses direction after reaching their final stop, which is about 10 minutes distance from NEC. Figure 4 illustrates this behavior in the alternating, long and short intervals between the probe requests. (A more detailed analysis – not shown for brevity – investigates stopping times and delays relative to the tram schedule.)

As in the previous example, output of the measurement tool helped pinpoint the cause of these periodic interferences with NEC's wireless network, caused by the local trams.

[1]The local tram operator is called HSB, for "Heidelberger Strassen- und Bergbahn AG."

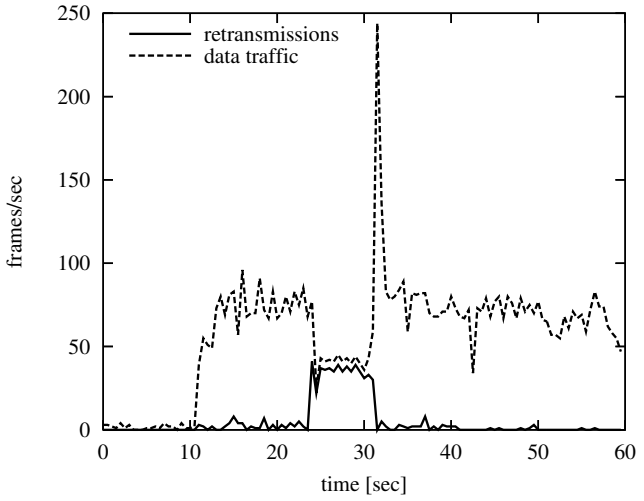

Fig. 5. Data frame and retransmission rates of a constant-bitrate UDP sender.

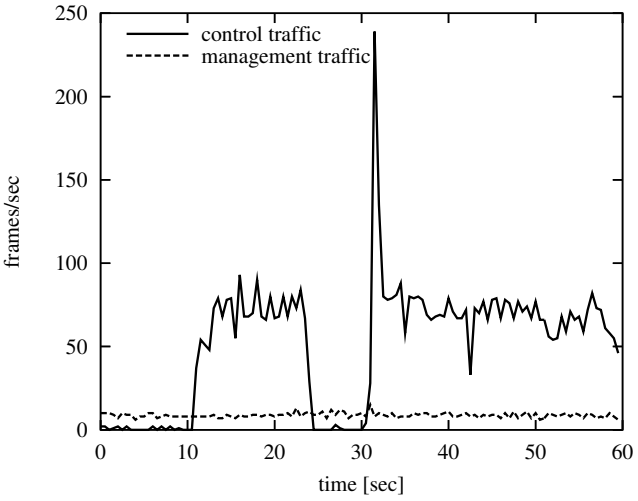

Fig. 6. Control and management frame rates of a constant-bitrate UDP receiver.

C. Detection of Connectivity Problems

This final example discusses how sequence number analysis can detect connectivity problems in wireless networks. As mentioned above, the measurement system deduces link-layer retransmissions based on observing repeated transmissions of *retry* frames with identical sequence numbers. Frequent retransmissions may indicate connectivity issues.

Figures 5, 6 and 7 illustrate different aspects of the same, one-minute traffic trace between a constant-bitrate UDP sender and its receiver.

Figures 5 shows the data frame and retransmission rate of the UDP sender. It starts transmitting ten seconds into the measurement. The average data rate of the stream is around 75 frames/second until second 24, when the data rate suddenly drops to about half for the next eight seconds before resuming at the original rate. This drop in the data rate goes along with a corresponding increase in the retransmission rate from second 24-32, also shown in Figures 5.

Analysis of the receiver-generated acknowledgment traffic in Figure 6 indicates that during the sender-observed connectivity problems (seconds 24-32), the flow of control traffic from the receiver to the sender almost completely stops. This explains the increase in retransmissions, because the sender repeatedly retransmits frames for which it receives no acknowledgments. However, Figure 6 also illustrates that management traffic between the two stations is still flowing.

As in the previous examples, a sequence number analysis reveals additional information. As Figure 7 illustrates, the rate of increase of the sender's sequence numbers is steady until 24 seconds into the trace, at which point the slope abruptly flattens before resuming the original rate at second 32.

The receiver's sequence number plot is more interesting. It shows that at time 32, the sequence number counter resets and restarts at zero. As mentioned above, this only occurs if the network interface is reset. Consequently, a malfunctioning receiver caused the connectivity problems in this example, either due to hardware failure or driver instabilities.

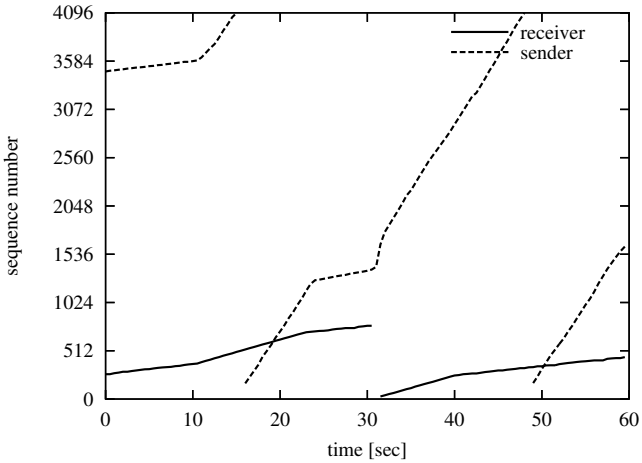

Fig. 7. Sequence numbers plots of a constant-bitrate UDP sender and receiver.

V. Conclusion

This paper presented new measurement tools that were motivated by the need to manage and troubleshoot NEC's local wireless networks. The goal was to design tools that can automatically determine common causes of problems based on analyzing ongoing network measurements.

One new measurement mechanism detects transient congestion or interference by observing link-layer frame retransmissions. Another mechanisms analyzes the link-layer sequence numbers of traffic traces and can, for example, more reliably detect spoofed frames. Additionally, these tools can aid a human operator during more complicated investigations by providing more meaningful metrics than raw traffic traces. One such metric are sequence number plots, which illustrate transmission rates over time and can help identify faulty hardware.

The paper illustrates the effectiveness of the new metrics and the measurement system through a series of three real-world experiments.

ACKNOWLEDGMENT

The authors would like to thank Enrico Giakas, Ralf Schmitz, Stefan Schmid, Xavier Pérez Costa and Martin Stiemerling for numerous suggestions that improved this work; the latter also for letting one of the authors continue this work after switching projects.

Lars Eggert was in part funded by the Ambient Networks project, partially supported by the European Commission under its Sixth Framework Program. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Ambient Networks project or the European Commission.

REFERENCES

[1] M. Roesch. Snort: Lightweight Intrusion Detection for Networks. Proc. *13th USENIX Conference on System Administration (LISA)*, Seattle, WA, USA, November 7–12, 1999, pp. 229–238.

[2] N. Provos. A Virtual Honeypot Framework. Proc. *13th USENIX Security Symposium*, San Diego, CA, USA, August 9–13, 2004, pp. 1–14.

[3] ANSI/IEEE Standard 802.11. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *ANSI/IEEE Standard*, 1999.

[4] 802.11 Handbook. A Designer's Companion. *IEEE Press*, 1999

[5] IEEE Organizationally Unique Identifier (OUI) List. December 2004. `http://standards.ieee.org/regauth/oui/oui.txt`

[6] J. Wright. Detecting Wireless LAN MAC Address Spoofing. *White Paper*, January 2003.

[7] J. Wright. Layer 2 Analysis of WLAN Discovery Applications for Intrusion Detection. *White Paper*, August 2002.

[8] Y. Lim, T. Schmoyer, J. Levin and H.L. Owen. Wireless Intrusion Detection and Response. Proc. *IEEE Workshop on Information Assurance*, United States Military Academy, West Point, NY, USA, June 18–20, 2003, pp.68–75.

[9] J. Bellardo and S. Savage. 802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions. Proc. *USENIX Security Symposium*, Washington, D.C., USA, August 4–8. 2003, pp. 15–28.

[10] S. Fluhrer, I. Mantin and A. Shamir. Weaknesses in the key scheduling algorithm of RC4. Proc. *8th Annual Workshop on Selected Areas in Cryptography*, Toronto, Canada, August 16–17, 2001, pages 1–24.

[11] A. Orebaugh, G. Morris, E. Warnicke and G. Ramirez. Ethereal Packet Sniffing. *Syngress Publishing*, February 2004.

[12] V. Jacobson, C. Leres and S. McCanne. tcpdump. `http://www.tcpdump.org/`

[13] A. Rager. WEPcrack. `http://wepcrack.sourceforge.net/`

[14] M. Kershaw. Kismet. `http://www.kismetwireless.net/`

[15] M. Milner. NetStumbler. `http://www.netstumbler.org/`

[16] The Shmoo Group. Airsnort. `http://airsnort.shmoo.com/`

[17] AirMagnet INC. AirMagnet. `http://www.airmagnet.com/`

[18] AirDefense, Inc. AirDefense. `http://www.airdefense.net`

[19] T. Oetiker. RRDtool. `http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/`