5-2009

EVALUATING THE USE OF SNMP AS A WIRELESS NETWORK MONITORING TOOL FOR IEEE 802.11 WIRELESS NETWORKS

Robert Johnson
Clemson University, RBJ1128@Gmail.com

EVALUATING THE USE OF SNMP AS A WIRELESS NETWORK MONITORING TOOL FOR IEEE 802.11 WIRELESS NETWORKS

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Computer Science

---

by
Robert Bern Johnson
May 2009

Accepted by:
Dr. James Martin, Committee Chair
Dr. James M. Westall
Dr. Brian Malloy

ABSTRACT

The increasing popularity of wireless networks has led to instances of high utilization and congestion, some of which have resulted in an interruption of network service. A thorough understanding of how IEEE 802.11 wireless networks operate is crucial to predicting and preventing future interruptions. There have been many studies performed on wireless networks. Of those that have captured data from the wireless side, most have used a form of wireless network monitoring known as Vicinity Sniffing (wireless sniffing from a location that is physically close to an access point to be in the broadcast range) as the primary means of capturing data. We believe that with recent advancements, SNMP is now capable of producing reliable results that were previously unattainable. We were presented with several obstacles in our studies, most of which are beliefs that SNMP is inadequate for monitoring IEEE 802.11 wireless networks. The claim is that SNMP provides either aggregated statistics or instantaneous values, and that it cannot report data on a per-device level, which is often desired so that individual details of a network's performance may be analyzed. Although the data is aggregated over some length of time, recent advancements do in fact allow for per-device details. Because of this, we believe that these claims are no longer valid, and that they are hindering the use of a very versatile tool. This study is motivated by the iTiger project which is a research project located at Clemson University. A prototype system has been developed allowing fans attending home football games to interact with a set of web applications using 802.11 enabled smartphones. A driving requirement behind the work presented in this thesis was to develop a framework for monitoring and analyzing the underlying IEEE 802.11 network used by the iTiger system. The work presented is based on a set of controlled experiments conducted in the football stadium. The result of our study will be to show that the latest generation of wireless equipment can provide data that was once thought to be available only from wireless monitoring. Through our analysis, we will provide a proof-of-concept that SNMP is more capable than previously thought and that the results obtained from wireless networks are as accurate, and in some situations even more accurate, than those statistics acquired from using the techniques of Vicinity Sniffing.

DEDICATION

This paper is dedicated to my wife Ann. Without your support and encouragement, this would never have been possible.

ACKNOWLEDGMENTS

The study presented in this paper was conducted with the computing facilities at Clemson University. The majority of the analysis was made possible due to the hard work and dedication of the open source community. In addition, all of the graphical visualizations that are presented in this paper are a result of the contributions of Microsoft to the computing community.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

INTRODUCTION

As technology advances and computers become less expensive, the networking infrastructure grows as well. Wireless networks based on the IEEE 802.11 standard are rapidly becoming commonplace as a means for clients to connect to the Internet. With this, there is a crucial need to understand the characteristics of the wireless traffic as well as the wireless medium itself [1]. There have been many studies on networks in general, and although most of these studies were conducted on wired networks, some have been focused on the wireless portion of the network itself. Since 802.11 wireless networks use a wired link to connect the wireless access points (APs) of the network to the Internet, most wireless studies were actually performed on the wired portion of the network and/or combined with *Simple Network Management Protocol* (SNMP) logs [1 and 8]. The measurements from the wired point-of-view are able to provide accurate network traffic statistics for the data that was transmitted across the wired section. However, they are unable to identify the wireless medium characteristics because they cannot observe the actual 802.11 wireless packets in the air [2].

SNMP provides a means to analyze the network device logs and provide statistics regarding the 802.11 network form the AP point-of-view. However, this data is either aggregated or instantaneous information that is dependent upon the SNMP polling interval, which is usually on the order of several minutes (typically every 1 – 5 minutes) [1, 2, 3, 5, and 6]. Since significant events may occur in a wireless network between polling times, SNMP alone cannot provide the granularity that is often desired. This leads to the concept of *Wireless Monitoring* by use of sniffers, also referred to as V*icinity Sniffing* [1, 2, 6, 8, 9, and 13]. This type of passive monitoring lends itself very well to data gathering and statistical analysis. Several statistics may be simultaneously monitored while the process of collection operates independently from the remainder of the network, thus having no impact on the network performance. One of the most common statistics that is available using vicinity sniffing is the level of network congestion [9 and 13].

Wireless Monitoring, however, is not without its own issues. Varying wireless channel conditions have the capacity to lead to measurement loss. The monitoring device (the sniffer) must be physically located close to the access point that it is monitoring so that it is within the broadcast range of the AP. This is due to the physical characteristics of the wireless medium, and that a wireless signal is unable to travel the same distance as a signal on a wired link. This shortcoming means that multiple sniffers may need to be deployed to establish the level of coverage that is desired.

Originally, the goal was to develop a measurement tool/capability for Clemson University's iTiger project. We began with the following facts:

- The location on campus that was of the most interest was in the football stadium's West End Zone
    - The wireless infrastructure consisted of a set of 802.11 access points that included a Cisco Wireless Controller for management of the network

- The Clemson Computing and Information Technology (CCIT) department uses MRTG as one of their primary network monitoring and measurement tools
- The research community recommends the use of wireless monitoring as the technique for monitoring an 802.11 network for research

This directed our studies to evaluate a system that provided a centralized monitoring capability through SNMP and a system that also supported the needs of research. After examining several different pre-existing tools, it was discovered that a pre-built system was not readily available that met these criteria. We then decided to build our own set of tools and techniques that would facilitate the data analysis so that we were not limited to the capabilities that were provided by Cisco's WCS Management Console.

After we conducted a literature survey, we focused on the results and analysis from several studies that indicated that Vicinity Sniffing was a required technique for monitoring and measuring IEEE 802.11 wireless networks. We also noted that the research community generally agrees that the use of SNMP is not recommended for evaluating the performance of wireless networks. We thought that there was misconception with regard to the viability of SNMP and we needed to better understand the advantages and disadvantages of the two techniques. We found that the two methods are actually converging. To some degree, we can see this with the use of current equipment.

The purpose of this study was to gain a better understanding of the convergence of these techniques and provide a forecast of what may be expected over the next few years. In addition, we wished to provide a suggestion to iTiger regarding the methodology that should be used for network monitoring and measurements. A set of experiments were conducted on an experimental wireless testbed that is monitored by both Vicinity Sniffing and SNMP. The objective of the study is to examine the capabilities of a state-of-the-art wireless infrastructure, to identify the set of capabilities available through SNMP-based monitoring that were previously only possible through Vicinity Sniffing, and to establish a measurement technique that allows us to assess and compare the two individual monitoring techniques.

The research community recommends the use of wireless monitoring for effectively monitoring a wireless network. Through the evaluation presented in this thesis, we will demonstrate that SNMP is a viable tool for effectively monitoring a wireless network. Using controlled experiments, we obtain measurement data that allows us to correlate typical results obtained from Vicinity Sniffing to SNMP results obtained from a modern 802.11 system. We also show that previous claims regarding the level of detail obtainable with SNMP and the type of data that may be collected are no longer valid.

OVERVIEW

802.11 Protocol

The IEEE 802.11 protocol is designed to manage and reduce contention in the wireless communication medium in a fair manner [9]. The protocol uses an algorithm known as *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA). In CSMA, a station wishing to transmit must first listen to the channel for a predetermined amount of time to check for any activity on the channel [17]. If the channel is sensed as busy, the station has to defer its transmission for a specific amount of time known as the *Backoff Interval* and then tries to sense the medium again [9]. If the channel is sensed "idle" then the station is permitted to transmit the frame to the intended destination. Upon successful reception of the frame, the destination must then respond by sending an acknowledgement message back to the sender. If the sender does not receive an acknowledgement within a specified amount of time, it attempts to retransmit the frame [9]. Figure 1 shows the layout of a Wireless MAC Frame.

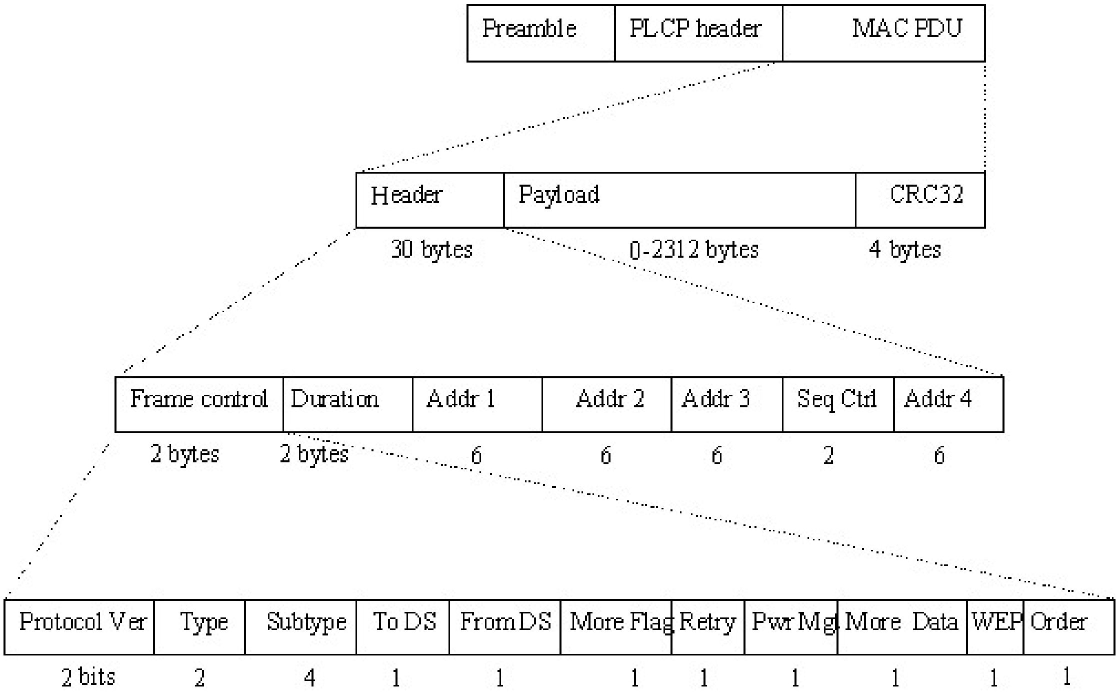


Figure 1: WLAN MAC Frame

Another method to reduce contention that may be employed is the use of *Request-To-Send* (RTS) and *Clear-To-Send* (CTS) messages between communicating hosts. A sender transmits a RTS packet containing all of the information regarding the size of the upcoming data frame and the amount of time that the channel will be occupied as required by the data frame. If the receiver is available to receive the data frame, it transmits the CTS packet not only back to the sender, but also to the other hosts so that the estimated channel-consumption time may be recorded [9]. The hosts that are not involved in the transmission then back off for the estimated amount of time, or until the channel becomes free again, before attempting to sense any traffic on the network medium.

In an attempt to increase the probability of successfully transmitted frames, many wireless card vendors employ a dynamically adjusted transmission rate algorithm known as *multi-rate adaptation* [9]. Multi-rate adaptation permits data frames to be transmitted at four different rates, 1 Mbps, 2 Mbps, 5.5 Mbps, and 11 Mbps. The IEEE 802.11 PHY protocol suggests that modulation schemes that are used for higher data rates (11 Mbps) are only able to tolerate low *bit error rates* (BER), whereas schemes used for lower rates (such as BPSK which uses 1 Mbps) can still function at a higher BER [13]. This implies that as error rates increase transmission rates decrease, and the reverse is true that as error rates decrease transmission rates increase. The disadvantage is that lower transmission rates result in lower network throughput [9].

SNMP

Simple Network Management Protocol (SNMP) is used in network management systems to monitor network-attached devices for conditions that require administrative attention. SNMP presents management data in the form of variables on the managed systems [18]. These variables may then be queried (and sometimes set) by managing applications. SNMP itself does not define which variables are accessible; Rather, SNMP uses an extensible design, where the available information is defined by *Management Information Bases* (MIBs) that are often proprietary to individual vendors. MIBs describe the structure of the management data of a device subsystem in a hierarchical namespace containing *Object Identifiers* (OID). Each OID identifies a variable that can be read or set via SNMP.

SNMP was introduced in 1988 to meet the growing need for a standard for managing *Internet Protocol* (IP) devices [18]. The main core of SNMP is a simple set of operations that provides the ability to query and set the state of some devices to network administrators. Although SNMP is capable of managing a wide variety of network devices (including but not limited to printers, personal computers, servers, power supplies, etc.), it is typically associated with routers and other network devices [18]. Just as with other protocols, SNMP is defined by The *Internet Engineering Task Force* (IETF) using *Request for Comments* (RFC) specifications. Currently, there are three versions of SNMP in use, and each one is defined by one or more RFCs. The RFCs that define each version are listed in Table 1.

SNMP Version	*Defining RFC(s)*
SNMPv1	RFC 1157 – Simple Network Management Protocol
SNMPv2	RFC 1905 – Protocol Operations for SNMPv2 RFC 1906 – Transport Mappings for SNMPv2 RFC 1907 – MIB for SNMPv2
SNMPv3	RFC 2571 – Architecture for SNMP Frameworks RFC 2572 – Message Processing and Dispatching RFC 2573 – SNMP Applications RFC 2574 – User-based Security Model RFC 2575 – View-based Access Control Model RFC 1905 – Protocol Operations for SNMPv2 RFC 1906 – Transport Mappings for SNMPv2 RFC 1907 – MIB for SNMPv2

Table 1: SNMP Versions and Their Defining RFCs

There is a standard MIB for IEEE 802.11 wireless networks, IEEE802dot11-MIB. Some of the OIDs defined within include AP configuration settings such as the authentication timeout and the interval with which beacons are transmitted. Also included are several metrics that are directly related to the overall performance of a wireless network like transmitted and received frame counts, successful RTS frames, as well as failed RTS frames for an individual AP [19]. In our study, we wanted to examine not only AP-specific data but also client-specific data. Cisco has extended the base 802.11 MIB with extensions that allow per-client events to be monitored. This extension is referred to as the AIRESPACE-WIRELESS-MIB [20]. Using this provided MIB, we are not only able to obtain the statistics that are included in the standard MIB, but we are also able to acquire client-specific statistics such as the number of packets and bytes sent and received, and even the SNR and RSSI values as seen by the clients and APs respectively [20]. Table 2 lists the metrics that were gathered from the AIRESPACE-WIRELESS-MIB.

Clients	*Access Points*
MAC Address	MAC Address of Wireless Interface
SSID Index Number	MAC Address of Wired Interface
Bytes Received Count	Name of AP
Bytes Sent Count	IP Address
Packets Received Count	Fragment Sent Count
Packets Sent Count	Fragment Received Count
IP Address	Frame Sent Count
Signal-to-Noise Ratio	Frame Failure Count
Average RSSI	FCS Error Count
MAC Address of Associated AP	Successful RTS Count
Number of Retransmissions	Failed RTS Count
-	Failed ACK Count
-	Number of Clients Connected
-	Receive Utilization
-	Transmission Utilization
-	Channel Number
-	Channel Utilization

Table 2: SNMP Metrics Gathered

WIRELESS MONITORING

Wireless Monitoring is a passive approach for capturing wireless-side traffic with rich MAC/PHY layer information [3]. It has been shown that Wireless Monitoring provides reliable analysis of the collected traces. Hence, it has recently been adopted in both wireless networking research and commercial Wireless Local Area Network (WLAN) management product development [1]. It is most commonly employed by the use of multiple computers often referred to as *sniffers*, each with a wireless network interface card (NIC). Each sniffer runs software that allows it to capture packets and analyze data, e.g., ethereal and tcpdump [1, 2, 3, 4, and 8]. Each sniffer is able to capture packets, but they are limited to the packets that are within range of their wireless NIC. Thus, each AP typically has its own sniffer [1 and 2].

Advantages

There are many advantages to Wireless Monitoring that are referenced by previous researchers, but the most commonly listed are the following [1, 2, 3, and 8]:

- A sniffing system can be easily designed and implemented
- Sniffing may be performed without any interaction with the existing network, and is therefore, completely independent of the operational network.

- Researchers are able to have more control over the geographical coverage of the measurement by simply adding or removing individual sniffers
- The sniffing devices theoretically can listen to any frames on the air within the range, so the data captures can be tailored to fit the needs of the researcher
- Sniffing can provide rich information on the wireless medium itself so that it is possible to infer the wireless medium characteristics such as physical and link layer information of each packet
    - This is not available if the traffic monitoring is performed in the wired part of the network
- Wireless sniffing allows for examination of physical layer header information including signal strength, noise level and data rate for individual packets
- When combined with timestamps, the collected data can be used as a good trace of the 802.11 link-level operations which can aid in protocol emulation or in problem diagnosis

Challenges

Conversely, along with the advantages of Wireless Monitoring are the common challenges that need to be addressed [1, 2, 3, and 8]:

- The advantages mentioned above would be invalid unless the sniffer can capture almost all of the packets in the air. Unfortunately it is very difficult to guarantee that the sniffers can see 100% of wireless frames
- It is difficult to accurately estimate the number of packets that were lost due to differences in wireless devices, drivers, antennae, etc
- Packet loss at the sniffer level is the most challenging problem in Wireless Monitoring. There are several categories of loss, frame loss, type loss and AP loss
    - Frame loss is the existence of such frames that are present on the air at the time of a measurement, but are not detected by the sniffer
    - Type loss is when the sniffer is inherently unable to capture specific types of packets
    - AP loss means that the sniffer loses nearly all the frames originating from specific APs.
- Typically most loss is due to signal strength variability, NIC variability, or a combination of both
    - For example, as a signal condition worsens, a sniffer is more susceptible to frame loss
    - Some specific NIC implementations do not permit ACK frames to be passed up to user applications, therefore resulting in type loss
    - AP loss may occur due to incompatibility between AP firmware and the NIC firmware

RELATED WORK

Many studies have been conducted on network monitoring. Most of them were based on wired networks because it is simple to obtain network traffic statistics in real time, or at least in near real time [3]. The problem with these studies is that they cannot be readily applied to wireless networks. To obtain the same level of detail in wireless networks as in wired networks, the previously noted challenges need to be addressed. The most common method that was identified was a controlled experiment, followed by an actual test of the system on a college campus with a functional 802.11 wireless network [1, 2, 3, and 8]. The analysis part of their work was conducted on tcpdump traces, rather than on a live system. These researchers mentioned above were trying to validate their Wireless Monitoring sniffing system. This section provides a brief overview of the studies that were examined. Only the main points that pertain to their objectives, methods, and key results will be highlighted.

Jardosh Et Al.

The first effort that is examined is that performed by Jardosh *et al.* and can be found in [9 and 13].

Objectives and Motivation

The motivation for the study in [9 and 13] was the fact that IEEE 802.11 wireless networks are becoming increasingly popular, but that there is general lack of knowledge when it comes to the operational details. A thorough understanding of the intricacies of wireless networks will allow for optimization of current and future IEEE 802.11 wireless networks. The main goal was to prove that as congestion in the network intensifies, smaller frames are more likely to be successfully transmitted and received; and that in a congested network, the use of high data rates and the transmission of fewer frames reduces per-frame channel occupancy and decreases medium contention respectively [13].

Methodology

The way that they conducted their research was to use a technique called *Vicinity Sniffing* [9 and 13]. This is where a laptop(s) is physically placed within the broadcast range of an access point and has a wireless interface capable of capturing the packets that are handled by that AP. In addition, the data capturing took place at the 62nd Internet Engineering Task Force (IETF) conference. The conference was held in Minneapolis, Minnesota from March 6, 2005 to March 11, 2005. It was attended by 1138 participants, most of whom used wireless laptops or other wireless devices [9 and 13].

Results

Jardosh et al. concluded that the analysis of heavily congested wireless networks is crucial for the robust operation of such networks [9]. They used the acquired network traces to analyze several things, including the impact of congestion on throughput, goodput, channel busy-time, the RTS-CTS mechanism, and various forms of delay [13]. They suggest that the use of lower data rates to transmit frames in the network significantly decreases the network throughput and goodput [9]. Even though most hardware manufactures program their network devices to use a lower data rate during times of congestion, this practice should be avoided in favor of higher data rates, even during periods of high congestion [9 and 13].

Cheng Et Al.

The next work that is examined is that performed by Cheng et al. in the paper "Automating Cross-Layer Diagnosis of Enterprise Wireless Networks".

Objectives and Motivation

The presence of IEEE 802.11 wireless networks is increasing, and becoming more prevalent in the workplace. Modern enterprise networks are of sufficient complexity that even simple faults can be difficult to diagnose — let alone transient outages or service degradations. Nowhere is this problem more apparent than in the 802.11-based wireless access networks now ubiquitous in the enterprise [14]. Not surprisingly, few organizations have the expertise, data, or tools to decompose the underlying problems and interactions responsible for transient outages or performance degradations [14]. The main objective of this study was to demonstrate techniques to infer the causes and effects of both link-layer delays and mobility management delays.

Methodology

As with most other studies, Cheng et al. performed their analysis on captured data. Their collection environment was the four-story UCSD Computer Science building [14]. The network that was monitored consisted of 40 APs covering all four floors, in addition to the basement [14]. The monitoring hardware consisted of 192 radios that were interspersed between the AP infrastructure. They also chose to use a form of vicinity sniffing in their study.

Results

Cheng et al. suggest that the analysis of network faults should be automated, and that networks must eventually address transient failure without the need for human involvement [14]. They have developed a set of models that take, as input, wireless trace data and can accurately determine the impact of protocol behavior from the physical layer to the transport layer on transmissions in the trace [14]. In addition, some types of delay can be directly measured, but many of the components must be inferred, such as queuing, back-offs, contention, etc. [14]. In the end, they determine that no one anomaly, failure, or interaction is solely responsible for these issues and that a holistic approach to the analysis may be necessary to encompass the range of problems that are experienced in real networks [14].

Raghavendra Et Al.

The study performed in [7] is the next to be analyzed. Raghavendra et al. were interested in the handoffs from one AP to another in a wireless network.

Objectives and Motivation

As wireless networks become more popular, they are being deployed in more and more varying conditions. At the same time, the monitoring of these networks has also advanced and continues to reveal key implementation deficiencies that need to be corrected in order to improve operation and end-to-end performance [7]. An increase in network load can give rise to several problems such as intermittent connectivity, low throughput, and high loss, resulting in an unreliable network, and in the worst-case, complete network failure [7]. The objective in this study was to analyze the impact of congestion and AP handoffs in an IEEE 802.11 wireless network.

Methodology

The data that was used for analysis in [7] came from the 67th IETF meeting held in November 2006. The network consisted of about 55 APs on both 802.11a and 802.11g networks, and was used by more than 1200 users over a span of five days, four of which were included in their traces [7]. They used the Vicinity Sniffing technique that consisted of 12 laptop-sniffers deployed at various locations. The sniffers were physically placed directly below the APs to maximize the likelihood of all packets being captured [7].

Results

Raghavendra et al. concluded that the handoff mechanisms need to be adaptive to congestion losses. The use of packet loss information to trigger handoffs resulted in a high rate of handoffs, even when the client was not mobile [7]. In the IETF network, a significant fraction of the recorded handoffs were to the same AP, and thus unnecessarily contributing to the network congestion. Further, many of the handoffs that occurred to other APs affected the clients negatively. Schemes that use signal strength trends to detect disconnection, and schemes that incorporate network information such as load in conjunction with loss are needed to avoid unnecessary handoffs [7].

Yeo Et Al.

Yeo et al. conducted the research outlined in [1, 2, 3, and 8]. They attempt to highlight the application of wireless monitoring.

Objectives and Motivation

As with the work in [9 and 13], the researchers in [1, 2, 3, and 8] stress the fact that a detailed understanding of wireless networks is needed. They also point out that most of the previously conducted research has been from the wired side of the network, and not conducted on the wireless portion itself [1, 2, 3, and 8]. The objective was to focus on implementing an effective wireless monitoring system and demonstrating its effectiveness in traffic characterization and network diagnosis [1].

Methodology

The analysis in [1, 2, 3, and 8] was conducted on a set of data that was obtained from a number of controlled experiments. For the pitfalls that they identified, a feasible solution was proposed and implemented to create a reliable wireless monitoring system [1, 3, and 8]. After determining the effectiveness of their system, it was applied to a real WLAN network in the Computer Science department of a University [1]. The captures were allowed to run for a period of two weeks. Then, they demonstrated how the captured data might be effectively used for both traffic characterization and network diagnosis [1, 2, and 3].

Results

Yeo et al. concluded that they identified the common pitfalls of wireless monitoring and that they provide two feasible solutions, merging data from multiple sniffers and the placement of the sniffers [1, 2, 3, and 8]. Those techniques were then applied to an academic research WLAN over a period of two weeks. They revealed not only typical traffic characteristics, but also some of the anomalies in the Media Access Control (MAC) protocol such as retransmission of some Management frames [1, 2, 3, and 8].

Analysis

The basic idea that remains constant throughout the research is that of wireless monitoring. The researchers simply need to place sniffers, most often a laptop, in the range of an access point and try to capture all of the data that they can. From that point, the studies tend to go down different paths.

The work done in [9 and 13] mainly focused on congestion and the rates at which data is transmitted. In [7], Raghavendra examined how network congestion leads to more AP handoffs, which in turn, leads to more congestion. Cheng et al. suggest that the analysis of network faults should be done entirely by computers, without any involvement from humans [14]. In addition, the work done in [1, 2, 3, and 8] focused on creating models of traffic for use by other researchers.

From the research conducted in [9 and 13], it is more apparent that the IEEE 802.11 wireless protocol actually hinders itself by using lower data transfer rates when the network experiences high levels of congestion. Due to the lower transfer rate, packets occupy more bandwidth than they otherwise would if they were transmitted at a high rate.

It is this greater amount of time that allows the packets to be affected by congestion [9 and 13].

Similarly, Raghavendra et al. shows another weakness of the protocol. When a wireless network experiences high levels of congestion, handoffs may occur in an attempt to alleviate the workload of the APs [7]. The process involved in a handoff actually contributes to the network congestion, and therefore has a negative impact on the overall network performance [7].

As previously stated, the common thread among all of the presented research is that of Vicinity Sniffing. While this is a proven method for network measurement, the use of SNMP would have provided further depth and insight in the analysis. In [9 and 13] the researchers focused on the rate at which data was transmitted across the network. Not all of the network traffic is guaranteed to be captured when using Vicinity Sniffing, and the amount of data captured varies greatly with the location of the sniffer(s). SNMP, on the other hand, is guaranteed to report statistics on every packet that crosses the network. As we will show, there is a great deal of variability in the network captures when it comes to the placement of the sniffers. One technique for dealing with this variability is to use a tool that reports on 100% of the network traffic like SNMP.

In the work found in [14], Cheng et al. state that network performance characterization should be done entirely by computers without any human intervention. While this would be ideal in some situations, but most of the time, a person is required to identify trends and make a decision as to whether or not two events are similar enough to be correlated as one. The more information, and accurate information, that is available to aid in this type of decision, the better the decision will be. We will show that although SNMP is still quite limited in its ability, it is now able to report some of the same metrics as Vicinity Sniffing, and it is able to report on these metrics from a more accurate measurement base.

PROBLEM STATEMENT

The problem addressed by this thesis is described by the following points:

- The majority of previous research has discounted the use of SNMP on wireless networks.
    - SNMP is not typically viewed as an accurate and reliable tool since it presents either aggregate or instantaneous information at the time it is queried.
- Vicinity Sniffing is still seen as the best method of observing a network's performance because of the level of granularity that is available.
    - When using Vicinity Sniffing in ideal conditions, it is possible to view the entire contents of packets that pass across the wireless network.
- Although Vicinity Sniffing provides for packet-level detail, some limitations exist that reduce the level of accuracy such as the physical characteristics of the wireless medium itself.
- SNMP does not provide the same level of detail as Vicinity Sniffing, but the data that is returned from a SNMP query covers 100% of the packets that have passed through a particular segment of the network, whereas the data obtained from Vicinity Sniffing is dependent upon the physical placement of the sniffers.

This thesis is designed to show that the latest wireless equipment is capable of producing results that are on a per-device level, and that the data retrieved from SNMP is valid, even for wireless networks. Further, we correlate data obtained through Vicinity Sniffing to data obtained from SNMP. This serves to partially validate that the results from both methods are accurate and to develop additional performance analysis dimensions that can be achieved when using an analysis methodology that includes both techniques.

THE PROPOSED METHODOLOGY

The proposed methodology is designed to aid in the process of data acquisition and analysis of network performance for IEEE 802.11 wireless networks for use in Clemson University's iTiger network. It is intended to evaluate the use and accuracy of SNMP as a measurement and monitoring tool for use on IEEE 802.11 wireless networks. It is not designed to become the sole source of information; rather, it is designed to provide an additional option for wireless network monitoring and measuring.

Background

Currently, Vicinity Sniffing is the most widely used method for wireless network monitoring. Although the use of SNMP has been explored in previous research, the general assumptions against it have stood the test of time. The primary reason (as given by previous research) not to use SNMP is that it presents aggregated or instantaneous statistics that are not updated with a desired frequency – by default, SNMP statistics are updated every one to five minutes [1, 2, 3, 5, and 6].

Motivation

Several factors contributed to the design and conception of this system. According to [2], it is observed that SNMP statistics cannot reveal per client information.

The authors also state that only wireless monitoring (Vicinity Sniffing) can capture the retransmission information per client. Due to recent advancements in wireless equipment, neither of these statements is true.

Recent advances in equipment do indeed allow statistics to be gathered that are client-specific. The SNMP table *bsnMobileStationTable*[1] lists statistics that are specific to individual clients, and may only be obtained from the individual clients themselves. Such information includes the number of packets/bytes sent and received, the *Signal-To-Noise Ratio* (SNR), the *Received Signal Strength Indication* (RSSI) from the client to the access point, and the current level of encryption that is supported and employed by each client [20].

Another contributing factor involves the advancement of networking technologies and management tools. Cisco's Wireless Controllers and Lightweight Access Points provide a device management structure in a hierarchical design. The device combination permits a network administrator to manage only one device, the controller. The controller automatically manages the other devices, like the Lightweight Access Points. This technology is then further developed by incorporating analysis tools. Cisco's Spectrum Expert [22] is one of these tools that have been developed to operate in this newly structured network design. The tool is able to integrate with the WCS Management Console that is used to manage the wireless controllers and provide charts for channel utilization and interference power. In addition, detailed summary statistics

[1] bsnMobileStationTable is part of the AIRESPACE-WIRELESS-MIB [20]

are available that include both wireless and non-wireless devices [22]. The contribution of Spectrum Expert by Cisco is an example of how vendors are extending their product lines with new capabilities that utilize the latest advancements in equipment.

Details

The proposed methodology offers support for the use of a tool that has not been heavily utilized in wireless networks, even though it has extensively used in wired networks. The data collection methods that are presented (paralleling those of the previous research) include the use of Vicinity Sniffing as well as gathering information via SNMP. This operation entails using the Vicinity Sniffing techniques, shell scripts, an SNMP client, the Perl scripting language, and a graphing tool such as Microsoft Excel[2] or Matlab. The shell scripts run the process by initiating an SNMP query and saving the entire output to a text file. Perl scripts are then run to parse the information and create CSV (Comma Separated Values) files that may then be opened by a graphing tool. After that is completed, the user may then create graphs and charts or use the data in another manner that is of their choosing.

[2] Microsoft Excel 2007 was used because it allows up to128k lines, whereas Excel 2003 only permits up to 64k [23].

Vicinity Sniffing

The decision to use Vicinity Sniffing was made because in order to analyze the full picture of the IEEE 802.11 wireless network, it is crucial to have traffic measurements from a wireless point of view [1 and 3]. In addition, the techniques of Vicinity Sniffing (frequently referred to by its generic name of *wireless monitoring*) are widely used in the research community, so much so that they have been completely relied upon in the past by several studies [8 and 13]. By using these techniques, granularity of the network statistics is possible down to the packet level.

Wireless monitoring has several distinct advantages over other techniques that make it useful for understanding the characteristics of traffic on a wireless network. Wireless monitoring systems may be set up and deployed without any interference to the existing network infrastructure [2]. The entire process is completely independent from the functionality of the underlying network. Due to the level of detail achievable by wireless monitoring, characteristics of the MAC/PHY levels may be inferred that are otherwise unavailable. It allows for examination of the physical layer header information as well as the link-layer header information [8]. Such information may be used to correlate with error rates and throughput to create models that are more accurate than those models that currently exist [2]. In addition, researchers may use the information to further develop, enhance, and optimize the IEEE 802.11 MAC protocol [21].

There are, however, some serious disadvantages to using wireless monitoring that must be taken into consideration. Without overcoming these hurdles, the integrity of the wireless monitoring techniques may be compromised. The two major issues are the accuracy of the network captures and the placement of the sniffers [1, 2, 3, 8, and 21]. A wireless monitoring system is not guaranteed to capture 100% of all traffic that crosses the network. In fact, the amount of data recorded is directly related to the physical location of the sniffers or monitors. A typical wireless monitor consists of a laptop computer with a wireless network interface card (NIC) that operates in Monitor Mode[3] and a protocol analyzer, such as Ethereal (Wireshark) or tcpdump, running to capture the frames [21]. One problem that often arises is the fact that not all NIC chipsets and drivers support RFMON mode.

SNMP

SNMP was chosen to serve as an additional data source to be compared to the use of the techniques of Vicinity Sniffing. Many experiments involving the use of SNMP for network monitoring have been conducted in the past, but they were performed on wired networks, or the researchers were primarily interested in aggregate network statistics and not the detailed packet-level network statistics that are available from Vicinity Sniffing [2 and 8]. The general assumption that SNMP cannot provide network statistics from a wireless client's point of view or from an access point's viewpoint has significantly hindered the widespread use of the tool.

[3] Also known as RF Monitoring (RFMON)

The advantages of using SNMP as a data acquisition tool are numerous and very distinct to certain uses and situations. The most important advantage of SNMP that made it popular is its simplicity in design and implementation [18]. The simple design easily allows users to identify variables or statistics that they want to monitor. The expandability of SNMP is also another advantage; because of its simple design, the protocol can be updated to meet future needs. Another advantage of SNMP is that it is widely implemented. Almost all major vendors of internetwork hardware, such as bridges and routers, design their products to support SNMP, making it very easy to implement.

Similar to Vicinity Sniffing, disadvantages exist for SNMP. SNMP has security flaws that can give network intruders access to the information carried along the network. Intruders could also potentially shut down some hosts. However, the SNMP version 2 has fixed some security issues regarding privacy of data, authentication, and access control [18]. Another disadvantage of SNMP is that its simple design means that the information it deals with is neither detailed nor well organized enough to deal with the expanding modern networking requirements [18]. Nevertheless, the benefits of the SNMP protocol far outweigh the negative characteristics, and when used in conjunction with the wireless monitoring techniques of Vicinity Sniffing, the resulting information is unmatched.

In 2005, Cisco acquired Airespace, Inc., a provider of Wireless Local Area Network (WLAN) systems [25]. Along with the company, Cisco acquired the products that were developed by Airespace, Inc.; of specific interest to us were a wireless controller and several Lightweight Access Points. The controller-access point combination formed a hierarchical network topology where the controller was the only human-managed device. Not only were the Lightweight Access Points managed by the controller, but all network traffic that crossed the wireless network also went to the controller. This design permits SNMP information to be collected by the wireless controller, creating a single point of contact for data retrieval.

We examined the available SNMP MIBs that pertain to our study and found that the MIB AIRESPACE-WIRELESS-MIB contains the majority of the statistics that we were interested in observing. More specifically, we found that this MIB also contained tables that were dedicated to statistics on a per-client level. This meant that the previous claims regarding the level of granularity that is possible with SNMP were no longer valid. We also noted that because of the hierarchical design of SNMP, we were able to query one OID and have a large tree of data returned.

Validation

In order to correlate Vicinity Sniffing and SNMP results on wireless networks, we conducted several small-scaled controlled experiments. The experiments were designed to duplicate the use of Vicinity Sniffing that was utilized in [1, 2, 3, and 8]. Once the experiments were conducted and the wireless monitoring techniques were applied, we obtained the results and performed the analysis. We found that our experiments yielded similar results to the experiments that we were duplicating. Only after that point did we examine the results from the SNMP data that was recorded for the duration of the experiments. We found that our SNMP-based data is similar to our Vicinity Sniffing data. In addition, we found that due to the problem of sniffer placement and measurement loss that is associated with Vicinity Sniffing, the SNMP data that we captured is actually a more accurate representation of the overall network traffic. The details of the experiments and their results are described below.

CONTROLLED EXPERIMENTS

In order to validate our results and ensure that we were using accurate data, we conducted several small-scaled experiments that duplicated the efforts found in [1, 2, 3, and 8]. Similar to our guides, we used the techniques of Vicinity Sniffing to capture the IEEE 802.11 wireless packets that were transmitted between the access points and the wireless clients. Unlike most of the previous research, we also captured SNMP information for the duration of each experiment. We believe that this additional data source will provide us with a perspective of the network that has previously been unexamined.

We conducted our experiments on a portion of the Clemson University's campus wireless network that is reserved for experimental use. Originally, we conducted four experiments and analyzed the results. We found that there were a few discrepancies in our data when we compared the two techniques, mainly the total number of packets that each technique correctly observed. This led us to perform an additional set of experiments to add to the validity of our results as well as our methods of obtaining the data. The additional set of experiments is detailed in Appendix B. After analyzing the data from the additional set of experiments, we found that our first set was invalid. This is due to a configuration problem with one of the sniffers that resulted in multiple channels being monitored instead of just one. The details of the original experiments are described below for completeness, but they should serve to highlight that one of the challenges associated with Vicinity Sniffing is that it can be simple to use it incorrectly.

Network Setup

To perform the experiments we utilized Clemson University's iTiger network[4]. The decision to use the network came from the fact that it is isolated from the rest of the campus-wide network already in place. Additionally, in its current state, it is primarily a research network and infrequently used. The testing equipment consisted of several Cisco AIR-LAP1242AG-A-K9 Lightweight Access Points, a Cisco AIR-WLC4402-50-K9 Wireless Controller, two laptop computers running CentOS[5] 5.2 with Cisco Aironet PCMCIA wireless NICs, and three Nokia N800 series Internet Tablets. The iTiger network is pictured in Figure 2. The Cisco Wireless Controller manages all aspects of the iTiger access points with the exception of handling the IEEE 802.11 Beacon frames. This allows for a level of simplicity in managing the access points that is not available when every access point requires individual configuration. The centralized management point also provides a single device to poll for SNMP-related information.

[4] Information about iTiger may be found at http://itiger.clemson.edu/

[5] "The **C**ommunity **Ent**erprise **O**perating **S**ystem" – Information is available at http://www.centos.org

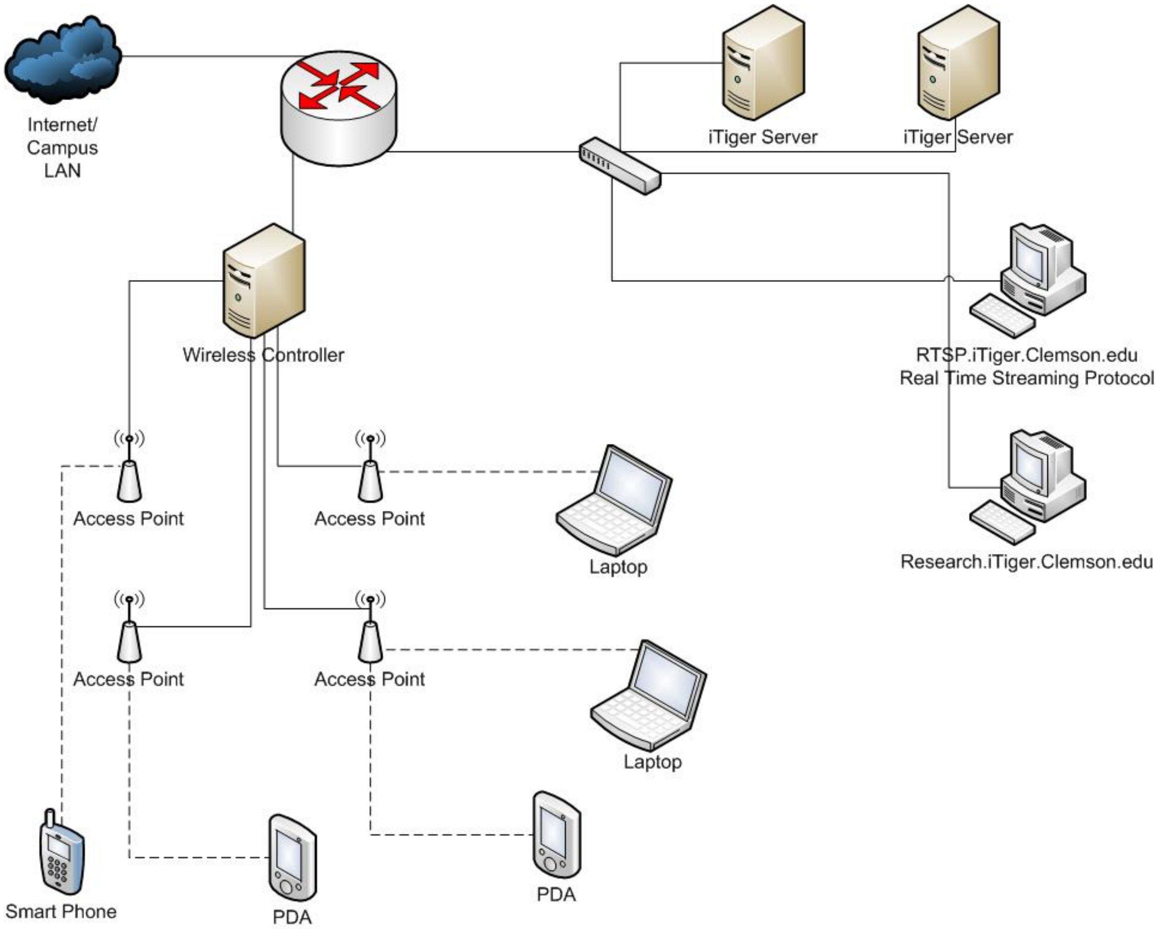


Figure 2: Clemson University's iTiger Network

Two of the three Nokia Tablets were N800s, and one was a N810. The N800s contain 128MB DDR RAM with an additional 256MB Flash ROM. They allow for connectivity to IEEE 802.11b/g wireless networks. The Nokia N810 Tablet has a 400 MHz processor, 128MB DDR RAM, 256MB Flash ROM, and an IEEE 802.11b/g wireless NIC. The two PCMCIA NICs were Cisco AIR-CB21AG-A-K9 Aironet

802.11a/b/g CardBus Adapters[6] that are compliant with the IEEE 802.11 wireless standards.

Methodology

The experiments that we conducted were not meant to be an exhaustive set; rather, they were designed to mimic those that were carried out in the past to validate the accuracy of the collected data. The validity of the SNMP-captured data depends on the validity of the experiments as a whole, so a strong foundation was required before the SNMP analysis could begin. As previously stated, four experiments were originally conducted, but another set was added to enhance the validity of our methods (see Appendix B).

With the exception of Experiment 1, the experiments were all similar to each other, only the placement of the clients and sniffers was changed. Experiment 1 was designed to serve as a baseline with only one client. Experiments 2 and 3 were organized in a way that highlighted the effect that sniffer placement has on the amount of data captured. Finally, Experiment 4 shows the effect of each client downloading the entire video twice. In all, the effect was that each client downloaded a 67MB file via HTTP. In addition, the access points involved were transmitting on channels one and eleven.

[6] Full specifications may be found at http://www.cisco.com/en/US/prod/collateral/wireless/ps6442/ps4555/ps5818/product_data_sheet09186a00801ebc29.html

Therefore, each client was also configured to use either channel one or eleven, and the two sniffers were set to capture data on those channels.

Table 3 lists each experiment configuration with the locations of the equipment that was used. A map of The West Zone Club at Clemson University, along with locations of interest, is depicted in Figure 3.

Experiment Number	***Sniffer Locations***	***Client Location(s)***
1[7]	8, 10	6
2	8, 10	3, 4, 6
3	8, 9	3, 4, 6
4	8, 9	2, 5, 7

Table 3: Experiment Configurations

[7] Only the Nokia N810 client was used in Experiment 1

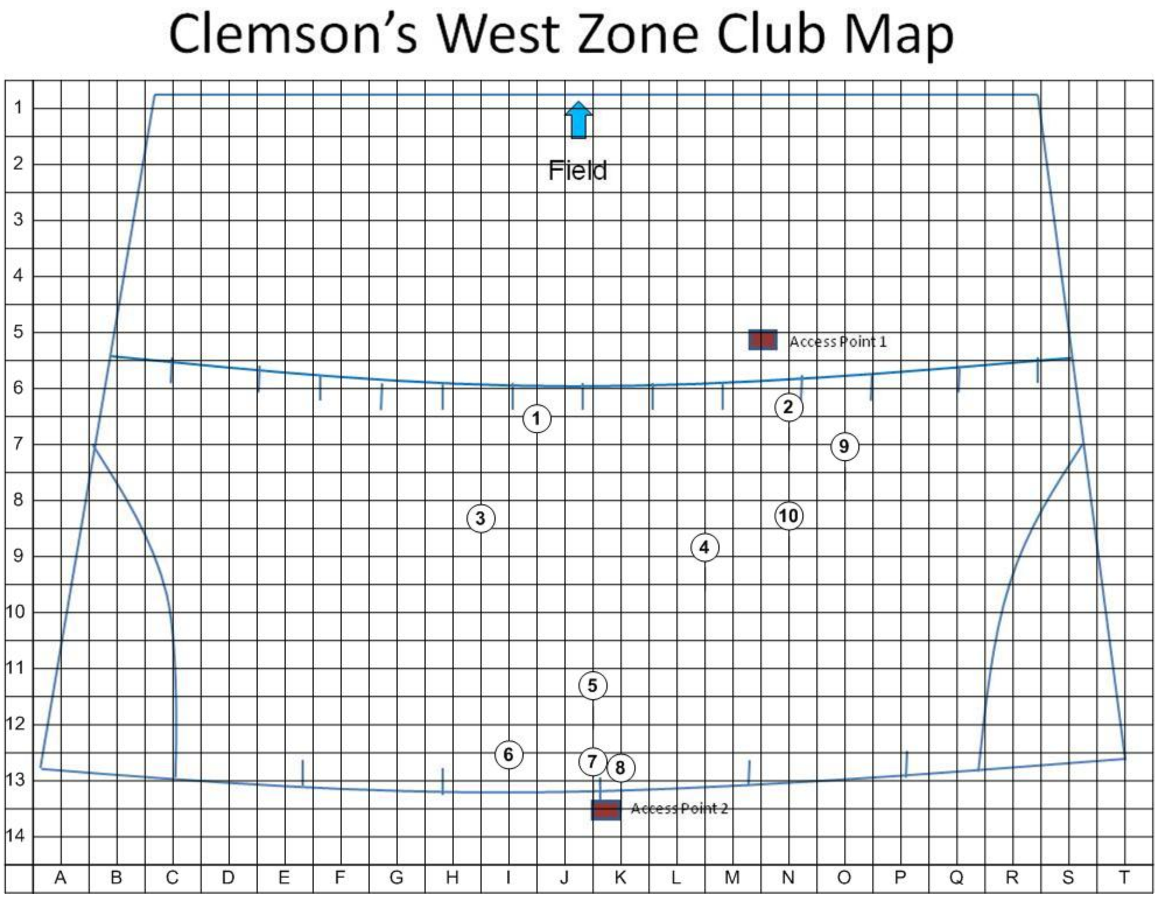


Figure 3: Map of Clemson University's West Zone Club

In each experiment, the wireless clients connected to the web interface of the iTiger *Real Time Streaming Protocol* (RTSP) server via HTTP (See Figure 2). Then, they each downloaded a video of a previously recorded football half-time show that was about fifteen minutes in length. This traffic was then captured by both sniffers concurrently via tcpdump. Similar to [1, 2, 3, and 8], we only captured the first 256 bytes of each wireless frame, but this is enough information to record the overall view of the frame (PHY/MAC/LLC/IP information) [1, 2, and 8]. Also captured was the IEEE 802.11 MAC frame header which includes (but is not limited to) the protocol version, type of frame, Source and Destination MAC addresses, and sequence number. Each sniffer created a separate network capture for each experiment. We then merged the captures from both sniffers to form one combined capture per experiment.

The process of collecting the SNMP data was automated by using scripts to issue the queries, save the output, and then parse the results. The computer labeled “Research.itiger.clemson.edu” (see Figure 2) was responsible for this. All SNMP data was acquired from the source labeled “Wireless Controller”. As previously stated, the use of the Cisco Wireless Controller with the Lightweight Access Points allowed us to query only one device and receive network statistics from all of the managed devices, including all of the wireless client devices. All of the SNMP data that pertains to the statistics that we were interested in could be retrieved from the Airespace MIB AIRESPACE-WIRELESS-MIB[8], so the scripts queried the wireless controller for all information that was under this OID.

On average, the data returned from this one query was 1.63 MB. The amount of information that was returned was more than what was needed, but the overhead of issuing individual queries soon became so large that it resulted in more network resources being consumed by SNMP than by data. This led to the decision to conserve network resources (at the expense of computer processing resources) and use one query, then parse the results.

[8] The AIRESPACE-WIRELESS-MIB translates to 1.3.6.1.4.1.14179.2

The data parsing was conducted by first searching for the SSID “itiger” to determine the set of clients that were of interest. Once the clients were identified, we then used their MAC addresses to search for the access points to which the clients were connected. By parsing the data in this manner, only the statistics that pertained to our experiments were included as output when the final data was written to CSV files.

RESULTS AND ANALYSIS

The methods used in this study were to reproduce previously published results and to compare them with our SNMP-based results. We show that SNMP can be more valuable than was previously thought and that the data gathered through SNMP is accurate. To better relate with the previously conducted research we performed wireless monitoring in parallel and compare the results and capabilities of the two methods. We begin by analyzing the data collected with the techniques of Vicinity Sniffing, and then proceed to analyze the data that was collected with SNMP. After the analysis, the two techniques are compared for their accuracy and level of detail.

Vicinity Sniffing

Table 4 provides a summary of the original set of experiments. It lists the number of packets that were recorded, the average bitrates, the average packet sizes, and the relative start and stop times for each experiment. The times listed are all in seconds elapsed since the beginning of the first experiment. The recorded frames are those that were observed by both access points. It should be noted that the experiment labeled "All" is not a sum of the statistics of the individual experiments. It is, however, the result of merging[9] the individual capture files from each experiment into one larger file before the analysis is conducted. This produces a capture file, and the subsequent statistical

[9] The capture files were merged with the tool *mergecap*, available from http://www.wireshark.org/

analysis, that encompasses the entire set of experiments. The statistics extracted from the combined experiment cannot be used directly for analysis since there are minor gaps in time between experiments. However, it does provide a summary view of all of the experiments.

Experiment	*Number of Packets Recorded* [10]	*Data Rate KB/sec.*	*Average Packet Size (Bytes)*	*Relative Start Time (Seconds)*	*Relative Stop Time (Seconds)*
1	353,366	36.9	159.00	0	1,490
2	578,865	136.2	279.01	1,684	2,841
3	276,496	36.5	164.16	2,936	4,152
4	1,236,866	221.1	317.72	4,288	6,024
All [11]	2,445,593	106.4	268.26	0	6,024

Table 4: Summary of Vicinity Sniffing Experiments

We present the results and analysis of the individual experiments as well as the combined experiments. The analysis of the combined experiments is mainly intended to serve as an overview with general statistics, whereas the individual experimental analysis provides details of a single experiment.

[10] The number of packets captured is dependent upon placement of the sniffers in relation to the access points, location of the clients, and physical barriers that prevent the sniffer from observing the traffic.

[11] Not the sum of the statistics of the individual experiments; rather, it is the result of merging the capture files from each experiment into one large capture file.

All Experiments Combined – An Overview

Figure 4 shows how the general data rate changes over time for all of the experiments. This behavior is consistent with what one would expect from analysis conducted on these types of experiments.

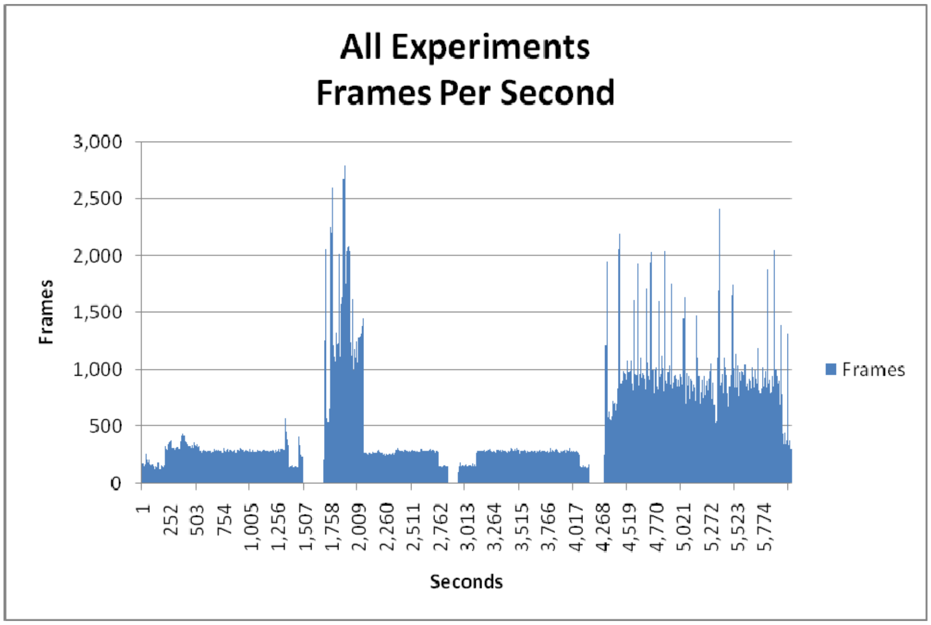


Figure 4: All Experiments – Frames per second

Figure 5 shows an overview of all experiments in bytes per second. Although the overall behavior of the traffic is the same, the analysis differs between frames per second and bytes per second simply because of the small size of the frames that are transmitted.

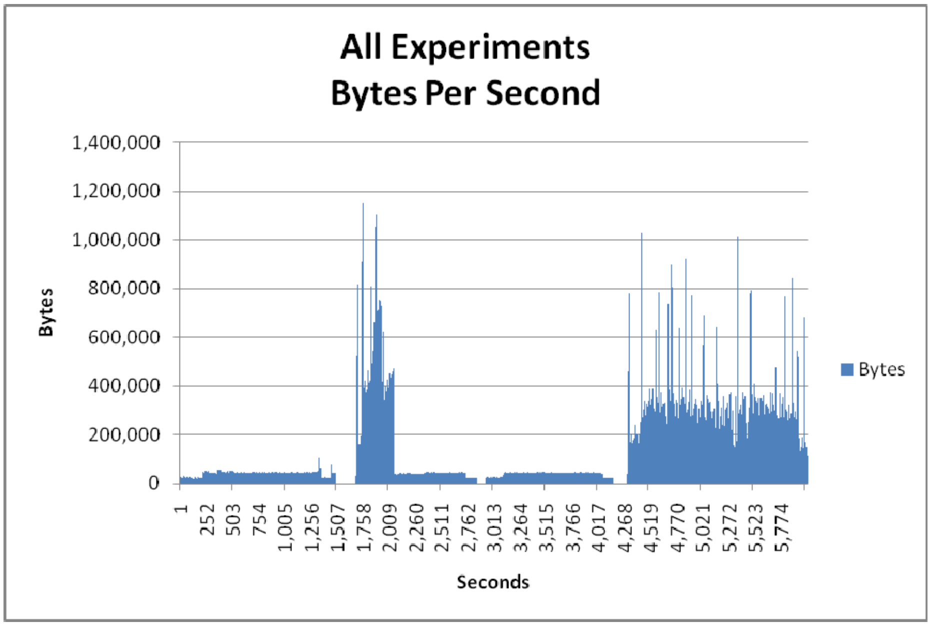


Figure 5: All Experiments – Bytes per second

The overall traffic characteristics of the frames are further broken down into their specific MAC frame types in Table 5, and presented graphically in Figure 6. We chose to analyze the MAC Management frames in addition to some Control frames, the same frames that were analyzed in the previous research. The largest percentage of the observed frames is IEEE 802.11 Beacon Frames. These statistics are also consistent with those found in [2]. It should be noted that there is a significant difference in the number of observed RTS frames and CTS frames. This difference may be attributed to the antenna characteristics of the NIC and the access points. An access point will typically contain an antenna that is capable of higher performance than a NIC. In addition, the close proximity of the sniffers and the access points meant that they were more likely to observe the CTS frames than the RTS frames.

Frame Type	***Count***	***Percentage***
Association Request	583	0.0281%
Association Response	338	0.0163%
Reassociation Request	459	0.0221%
Reassociation Response	407	0.0196%
Probe Request	77,953	3.7579%
Probe Response	82,677	3.9856%
Beacon Frame	1,127,477	54.3522%
Authentication	1,008	0.0486%
Deauthentication	1,079	0.0520%
Power Save Poll	944	0.0455%
Request to Send	12,396	0.5976%
Clear to Send	411,172	19.8213%
Acknowledgement	357,898	17.2532%

Table 5: All Experiments – Frame Types

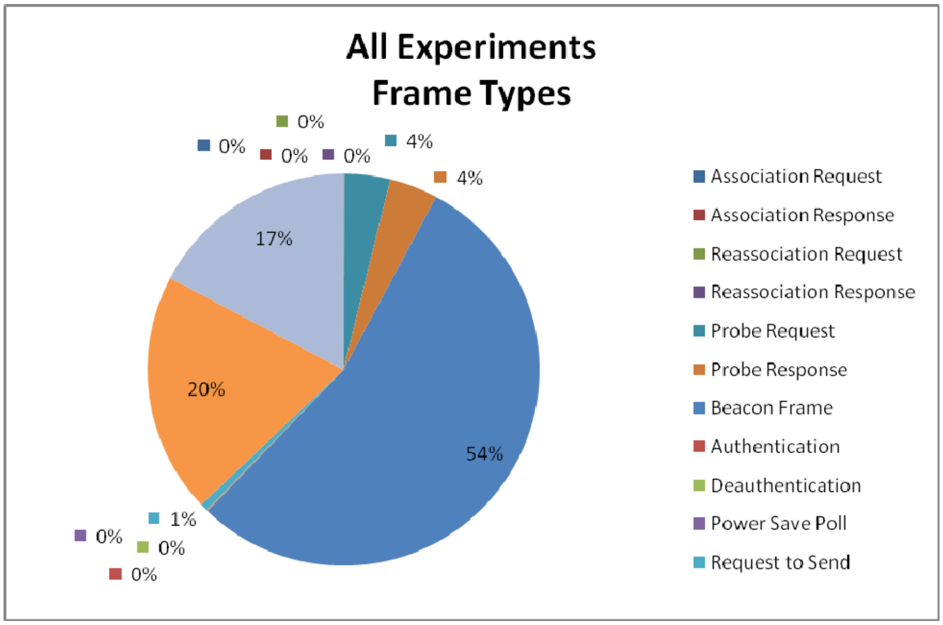


Figure 6: All Experiments – Frame types

Experiment 1

Figure 7 and Figure 8 show the throughput changes for the duration of Experiment 1 in frames per second and bytes per second, respectively. As previously mentioned, there was only one wireless client operating for the entire length of the experiment.

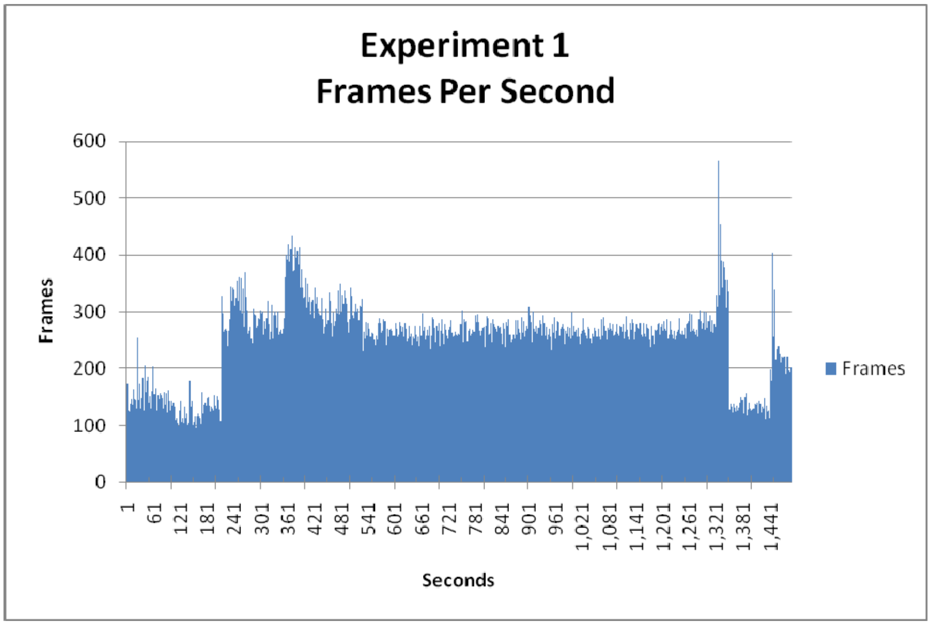


Figure 7: Experiment 1 – Frames per second

It can be seen on both graphs that there are spikes in the observed throughput. These spikes correspond to the time that the client was authenticating with the network, navigating to the iTiger website, and downloading the recorded video. In addition, towards the end of the experiment, it can be seen on the graphs where the video playback ended and the device was idle.

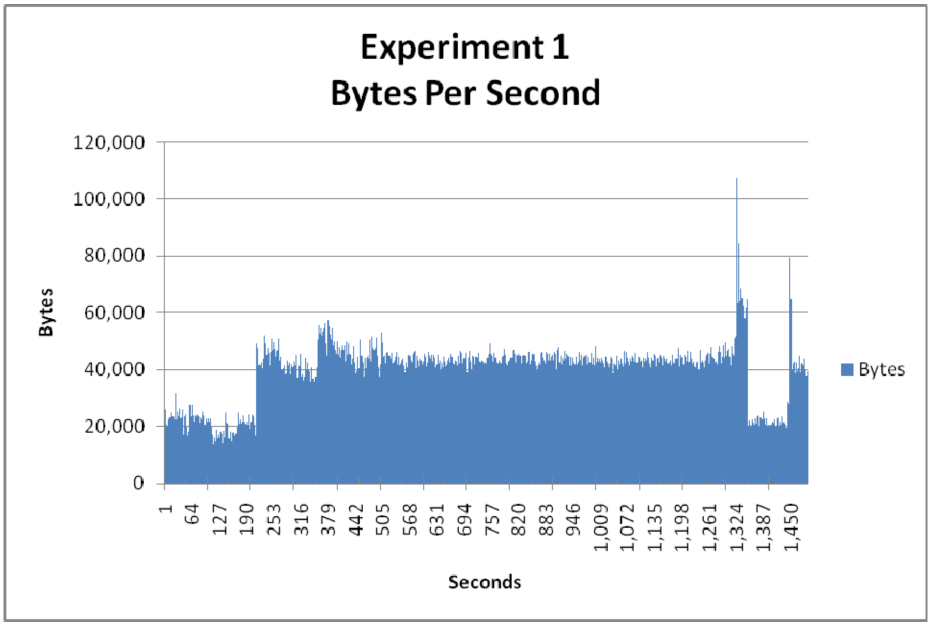


Figure 8: Experiment 1 – Bytes per second

The captured MAC frames for Experiment 1 are further detailed in Table 6. The majority of the frames are IEEE 802.11 Beacon Frames, followed by Probe Responses and Probe Requests. These findings are again, supported by the results found in [1, 2, 3, and 8]. The figures in the table are also visualized in Figure 9.

Frame Type	*Count*	*Percentage*
Association Request	161	0.0471%
Association Response	107	0.0313%
Reassociation Request	276	0.0807%
Reassociation Response	315	0.0921%
Probe Request	22,435	6.5600%
Probe Response	25,487	7.4524%
Beacon Frame	269,838	78.9007%
Authentication	854	0.2497%
Deauthentication	597	0.1746%
Power Save Poll	47	0.0137%
Request to Send	39	0.0114%
Clear to Send	1,425	0.4167%
Acknowledgement	20,416	5.9696%

Table 6: Experiment 1 – Frame Types

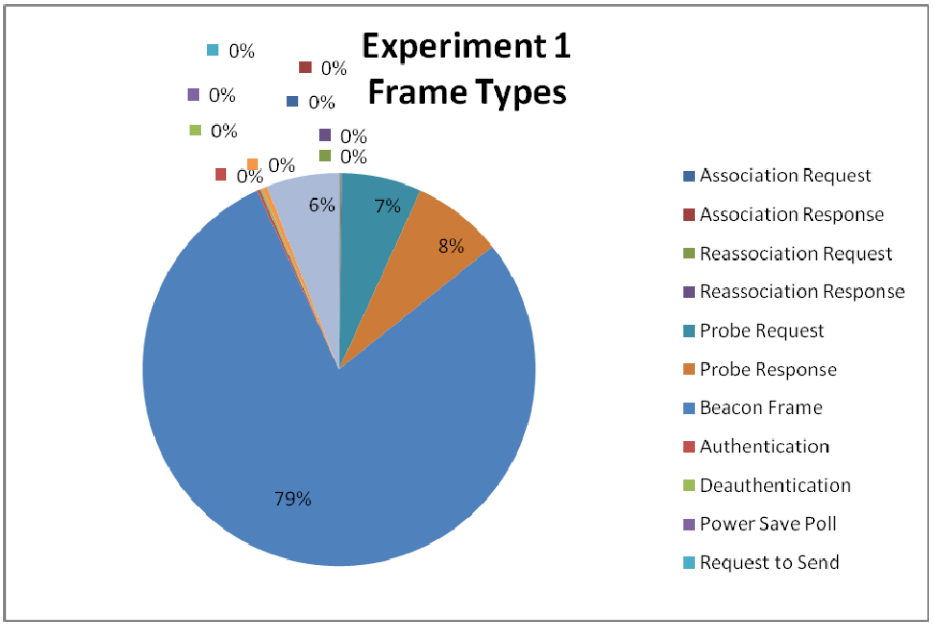


Figure 9: Experiment 1 – Frame Types

Experiment 3

In Experiment 3, the location of one of the sniffers changes from Experiment 2, along with the location of the clients. Referring to Table 3 and Figure 3, Experiment 2 had a sniffer at location 10, whereas a sniffer was moved closer to the access point (location number 9) in Experiments 3 and 4. The wireless clients were located at points 3, 4, and 6 for Experiment 2, and then moved to locations 1, 4, and 7 for Experiment 3.

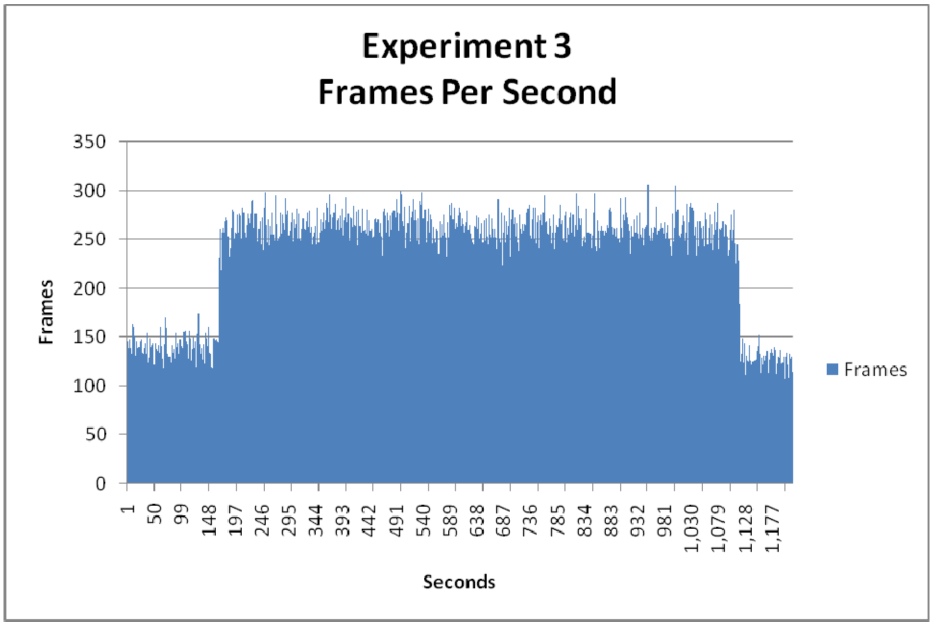


Figure 10: Experiment 3 – Frames per second

The effects of the new locations may be seen in both Figure 10 and Figure 11. The traffic itself did not change, but the clients and one of the sniffers moved closer to the access points. This change allowed the sniffers to capture more of the traffic than in Experiment 2. The fact that the amount of traffic that was captured changes based on the relative proximity of the sniffers, clients, and access points highlights one of the major challenges of Vicinity Sniffing that was mentioned above.

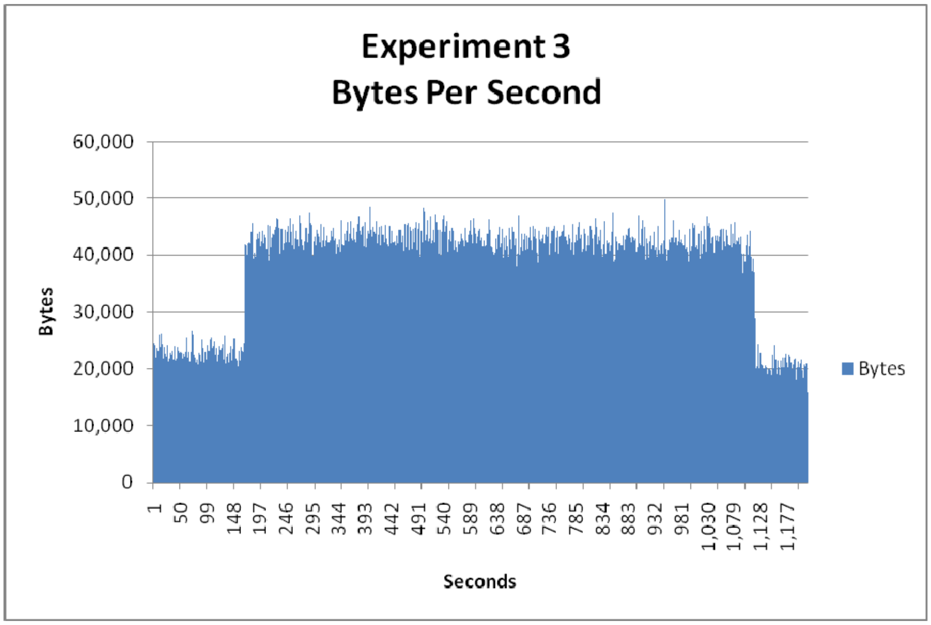


Figure 11: Experiment 3 – Bytes per second

Experiment 4

Experiment 4 is unlike any of the other experiments that were performed. For this test, each of the three clients downloaded the entire video twice[12]. The change in throughput is displayed below as frames per second in Figure 12, and in bytes per second in Figure 13. These graphs show that with a proper placement of the sniffers and the

[12] Since the Nokia devices do not maintain a cache for web content, each device downloaded the entire video each time it was viewed.

clients, the majority of the actual traffic is captured. Although the results from Experiment 4 are different from the others, they emphasize the fact that the accuracy of Vicinity Sniffing is highly dependant upon the relative locations of the wireless clients and sniffers to the access points.

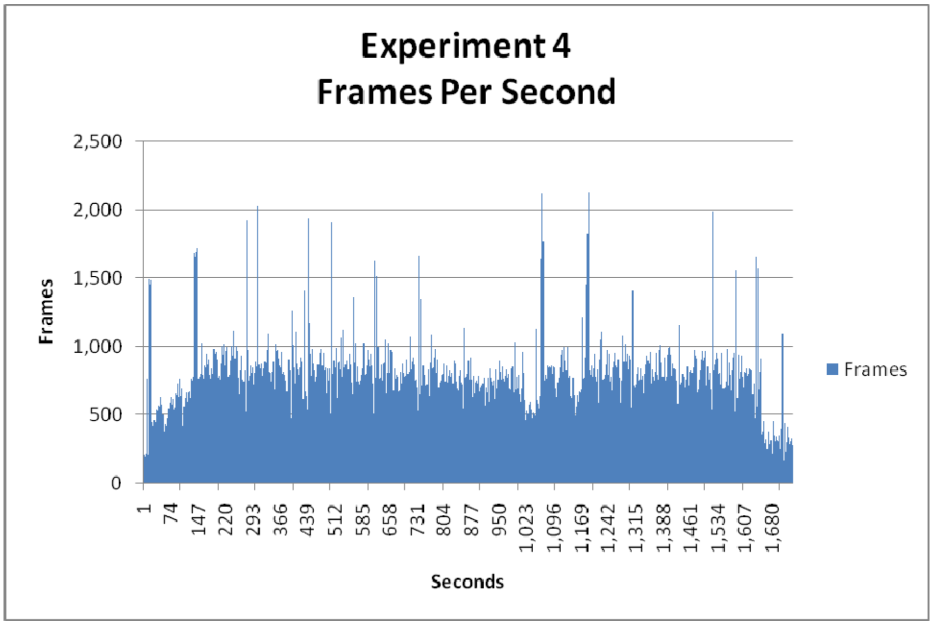


Figure 12: Experiment 4 – Frames per second

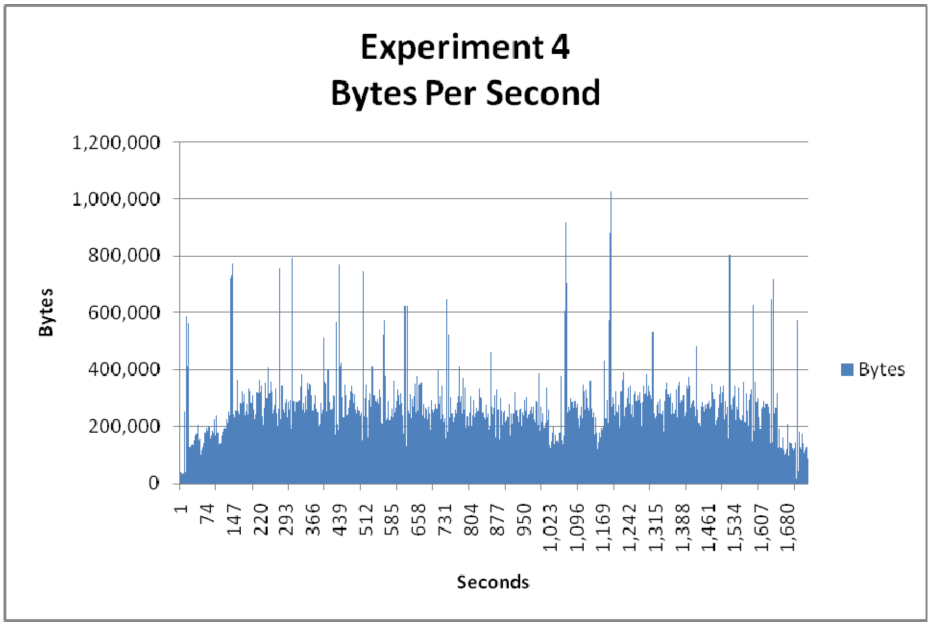


Figure 13: Experiment 4 – Bytes per second

Table 7 shows the details of the captured MAC frames, and they are graphically displayed in Figure 14. Similar to Experiment 2, the largest group of IEEE 802.11 MAC frames is Beacon Frames. This is again followed by Clear-to-Send frames and then Acknowledgement frames. These results are also consistent with those found in the previously mentioned work. Again, the difference in the RTS/CTS frame counts that was previously explained is clearly visible in the table.

Frame Type	*Count*	*Percentage*
Association Request	181	0.0184%
Association Response	106	0.0107%
Reassociation Request	110	0.0112%
Reassociation Response	45	0.0046%
Probe Request	23,259	2.3588%
Probe Response	24,989	2.5342%
Beacon Frame	414,103	41.9953%
Authentication	68	0.0069%
Deauthentication	273	0.0277%
Power Save Poll	751	0.0762%
Request to Send	9,842	0.9981%
Clear to Send	297,156	30.1354%
Acknowledgement	215,187	21.8227%

Table 7: Experiment 4 – Frame Types

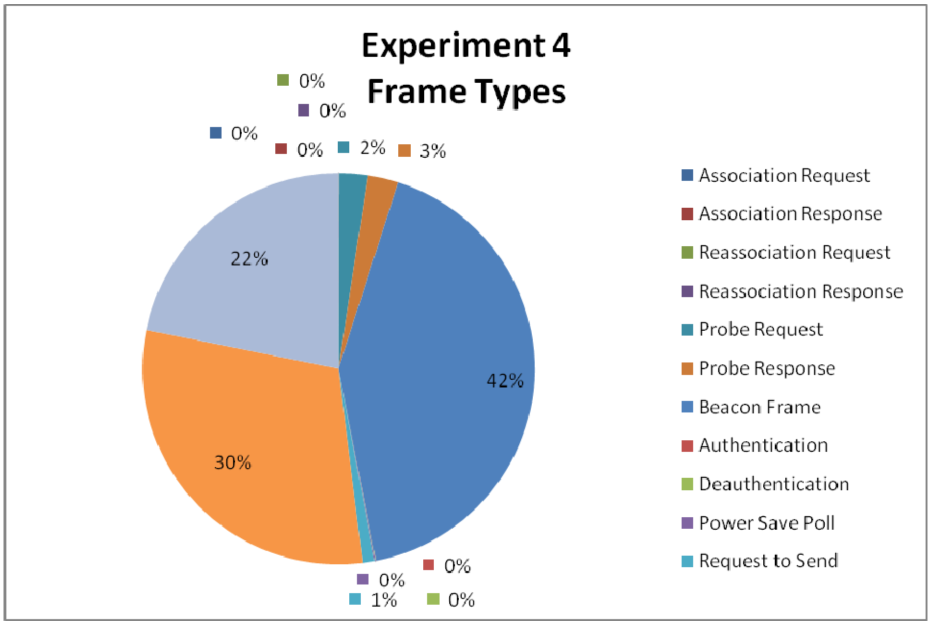


Figure 14: Experiment 4 – Frame Types

SNMP

Throughout the entire duration of the experiments that we performed, SNMP data was being collected by the computer labeled "Research.iTiger.Clemson.edu" (see Figure 2). We utilized the fact that the Cisco Wireless Controller managed all of the Lightweight Access Points, and allowed us to retrieve network statistics from one device. Had this not been the case, we would have been required to issue separate SNMP queries to every device. Another key point to mention is that the physical connection between the wireless controller and the SNMP-querying machine is a wired link. This means that since the average size of the returned data was about 1.63 MB, there was little negative impact from our queries.

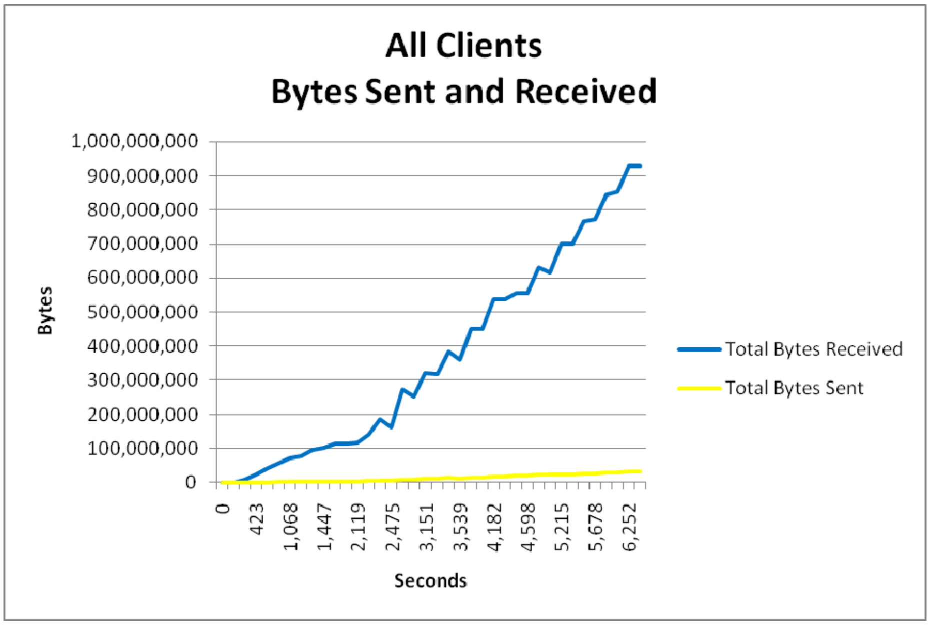


Figure 15: All Clients – Total Bytes Sent and Received

Figure 15 shows the total number of bytes sent and received from the perspective of the wireless clients, and Figure 16 shows the number of packets sent and received (from the clients' perspective as well). The data represents all three clients together. The figures are as one would expect; more data is received by the clients than is sent, and the total number of bytes/packets increases as time advances. One interesting point is that the total number of packets sent by the clients is similar to the total number of packets received. This is because the packets that are sent by the clients are much smaller than the packets that are received.

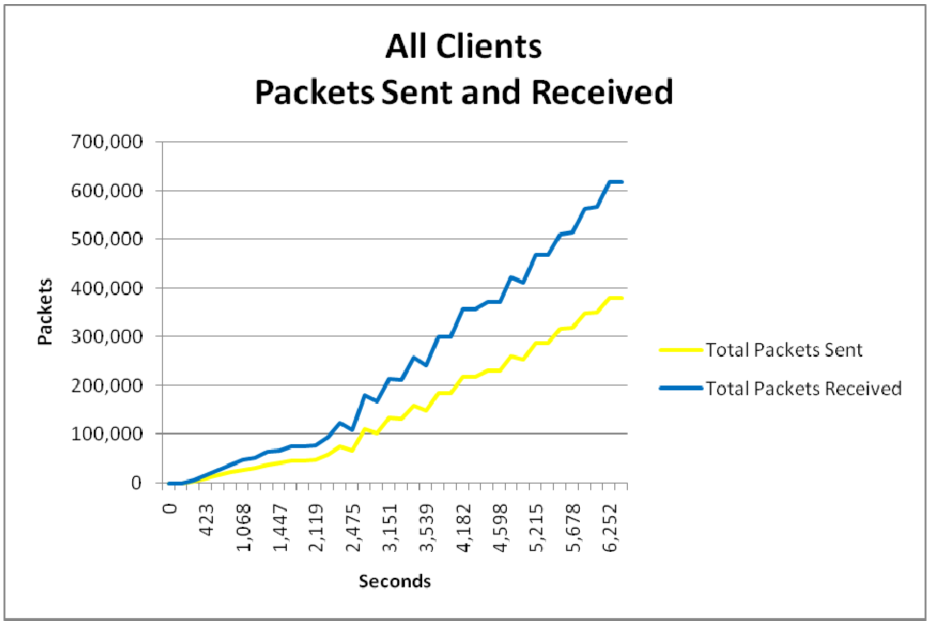


Figure 16: All Clients – Total Packets Sent and Received

The statistics from the perspective of the access points that we used are presented below. Figure 17 shows the total number of RTS failures and successful RTS frames for all of the access points at each time interval. Figure 18 shows the average utilization of all of the access points. The transmission utilization and the channel utilization closely follow each other. This is because the clients are downloading data and not uploading. One interesting observation is that the utilization is never over 18% even though there are no competing processes or data transfers. This data is presented as frames from both of the access points that were used.

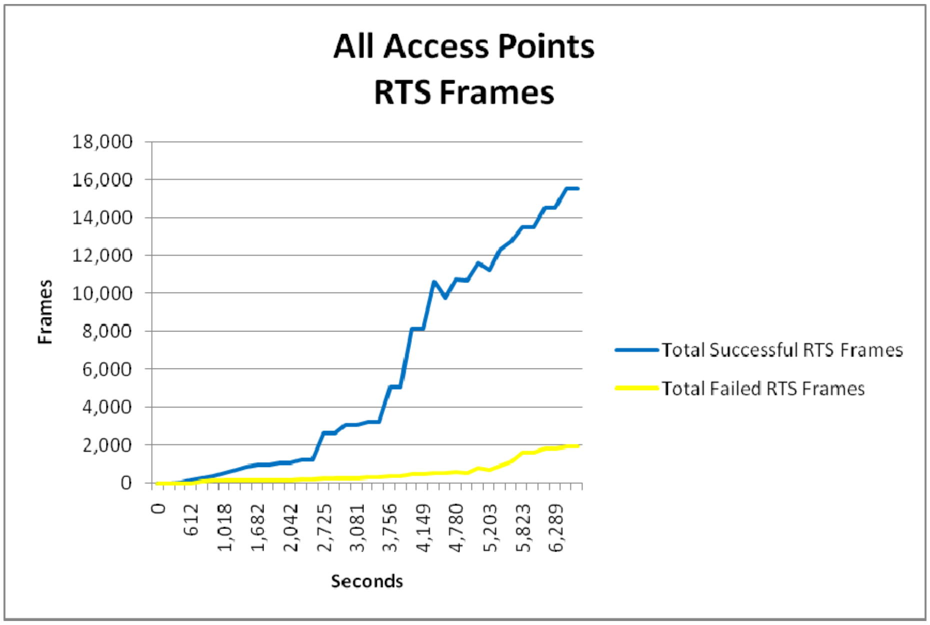


Figure 17: All Access Points – RTS Frames

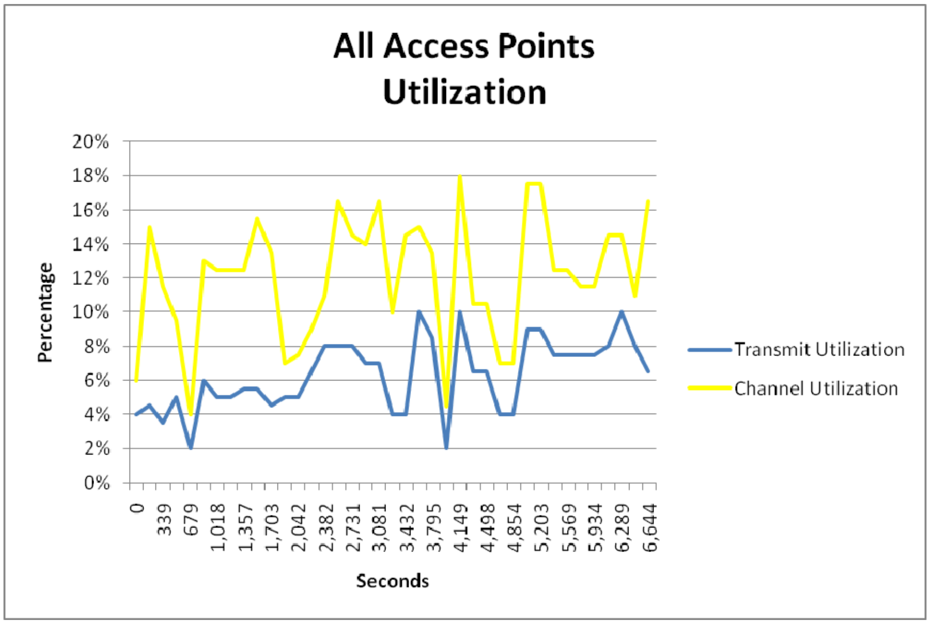


Figure 18: All Access Points – Utilization

The figures above presented the SNMP-based statistics in an aggregated form that encompasses all of the wireless clients (or the access points) that we used as a whole. The figures below are of the same SNMP-based statistics presented on a per-client and per-AP basis. This type of analysis highlights the usefulness of per-device statistics and negates previous claims that SNMP is not appropriate for performance analysis because it cannot show per-device information.

Figure 19 shows the number of packets that are sent and received by each wireless client, and Figure 20 displays the number of bytes that are received, also separated by each wireless client. Within the SNMP tables, the individual clients are identified by their MAC address, so the traffic that is presented below is also organized in a similar manner.

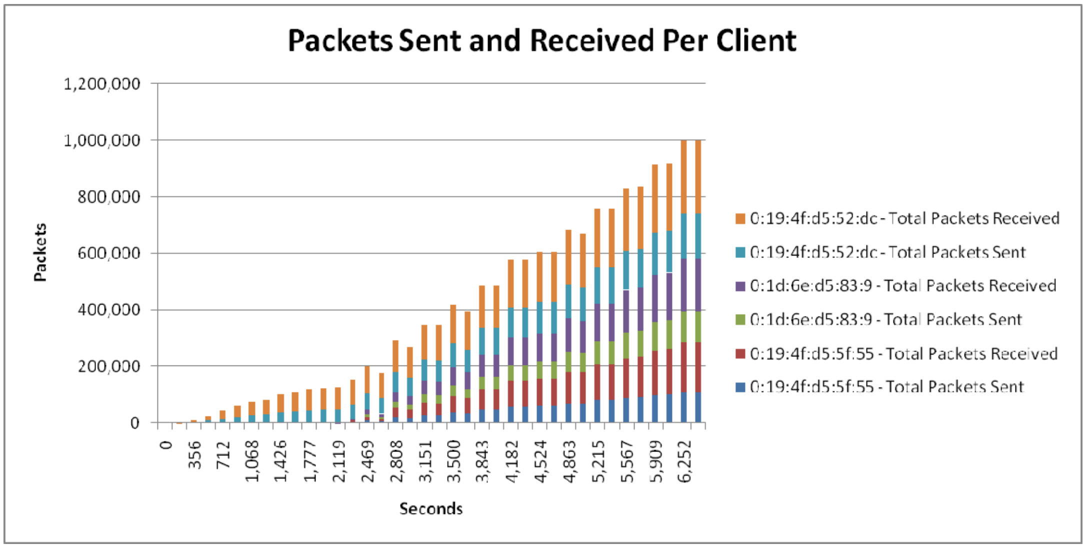


Figure 19: Packets Sent and Received Per Client

In examining these two graphs of the wireless client traffic, it is clear to see that initially, only one device was communicating. This corresponds to Experiment 1 in which there was only one active client. Similar to the above graphs, it appears that each client transmits and receives about the same number of packets, but in reality, not all packets are created equal; the size of the client's transmitted packets are much less than the size of the received packets, so much so that their contribution was negligible and therefore omitted.

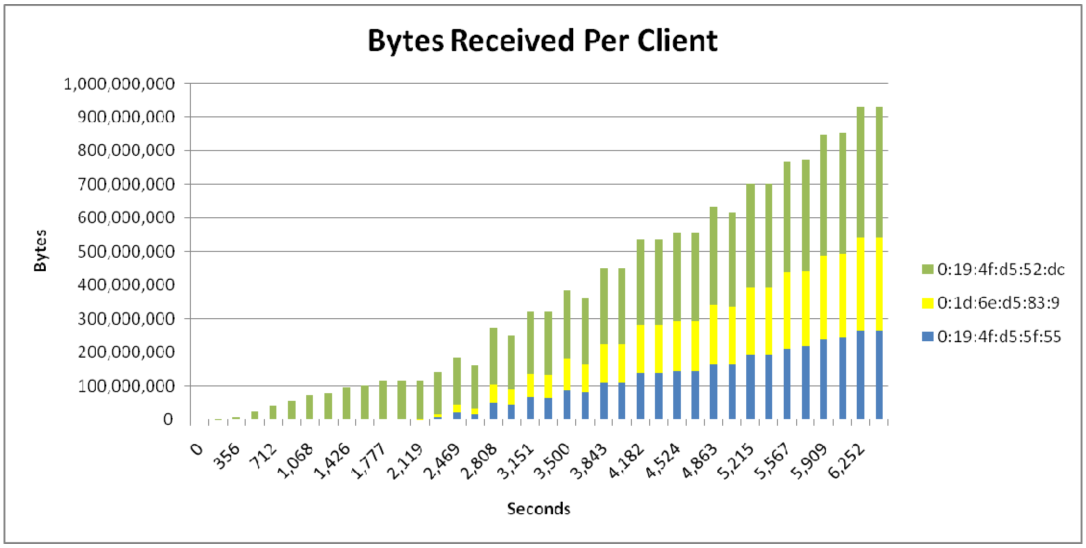


Figure 20: Bytes Received per Client

Figure 21 shows the average *Received Signal Strength Indication* (RSSI) of each wireless client that was used. Again, this graph displays the signal levels on a per-client basis. Just as with the above two graphs, it is clear to see when Experiment 1 ended and the other two devices came online.

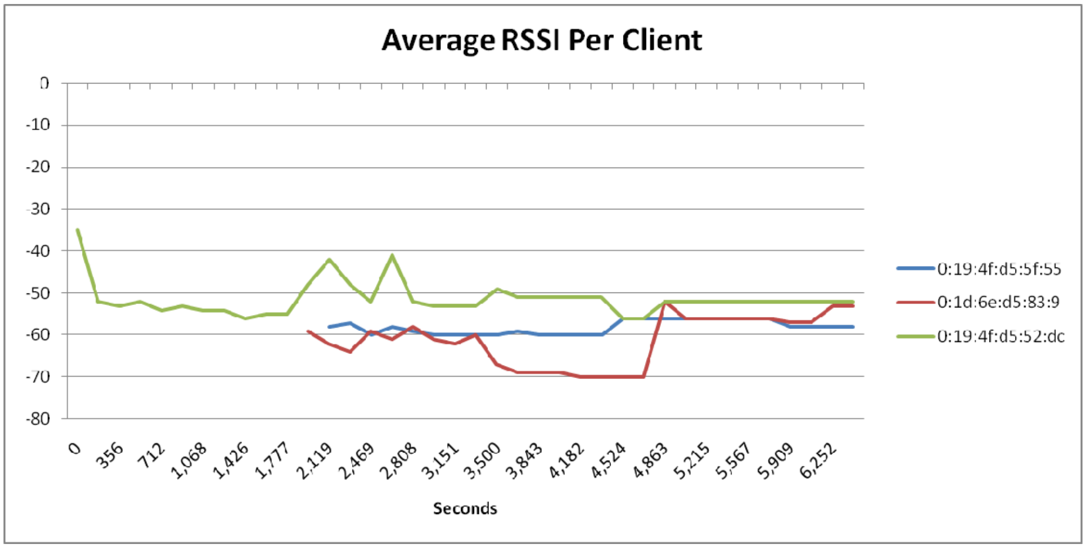


Figure 21: Average RSSI per Client

Figure 22 shows the same information as Figure 17, and like the previous three graphs, it displays the statistics on a per-device level. The RTS Frames are displayed per access point. The previous RTS graph only displays the overall total number of RTS frames, but the graph below shows more details. Without the below graph, it would not have been clear that one access point saw the largest number of successful RTS frames and another access point saw only a few successful RTS frames.

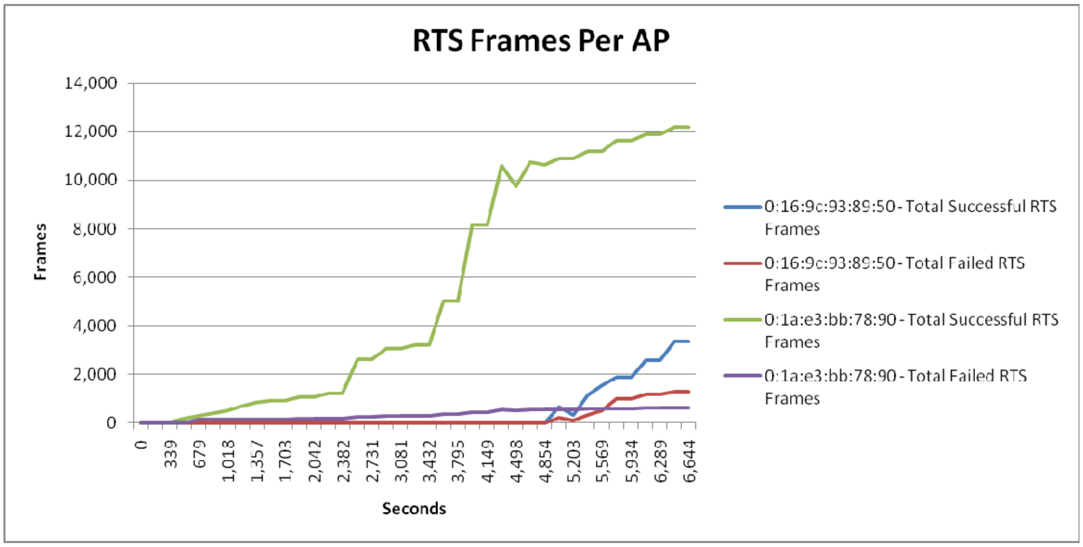


Figure 22: RTS Frames per Access Point

Analysis: Correlating the Results

When individually examined, both of the presented techniques for data acquisition produce useful and informative statistics. There is however, one caveat to using the aforementioned techniques; there is a tradeoff between the accuracy of the data and the level of detail that can be achieved.

Vicinity Sniffing is able to provide packet-level details, but it is not guaranteed to capture every packet. As noted above in the challenges of Vicinity Sniffing, the placement of the sniffers in relation to the access points greatly affects the quantity and quality of captured packets. This was illustrated in the experiments that we conducted. When the sniffers were relocated to new positions, the number of packets that were captured changed. The network traffic did not change from Experiment 2 to Experiment 3, but the results from both of those tests are different. This demonstrates that the accuracy of Vicinity Sniffing is highly sensitive to the placement of sniffers and the location of the clients in relation to the access points. Also highlighted is the fact that the sniffers may be incorrectly configured, and that the error may be difficult to detect.

As shown in Table 8, there is a difference between the total frame counts that were captured by each technique. Even though previous research has identified that Vicinity Sniffing is able to capture up to 99% of the traffic, the use of the Vicinity Sniffing techniques in our original experiments only captured about 64% of the traffic that SNMP was able to capture. This difference in accuracy can be attributed to several factors. First, the previous research was conducted over a much larger time span than our study. This means that their results were able to include small events, such as throughput spikes, in the averages but because of the large amount of data that they collected, the small events do not show up as profound as they do in our results. Second, our SNMP results are reported from the starting time of the first experiment to the finishing time of the last experiment, whereas the Vicinity Sniffing data is reported without the gaps of time that were required to setup and initiate each experiment. This means that during the time between Experiment 3 and Experiment 4, SNMP was still gathering data but the sniffers were not.

Third, our original experiments were conducted with the sniffers monitoring two different channels, although our guides only used one channel. This oversight was corrected in the additional experiments that are described in Appendix B. However, after examining the data that was collected from the additional set of experiments it was found that the Vicinity Sniffing throughput measurements from both sets of experiments are generally consistent with each other.

As shown above, the experiments that we ran all consisted of the same data, yet there were a different number of Vicinity Sniffing captured frames from each, whereas the SNMP data is reporting on all of the frames, regardless of whether or not a sniffer was within broadcast range.

Metric	***Vicinity Sniffing***	***SNMP***
Total Frames Captured	2,445,593	4,010,772

Table 8: Total Frames Captured By Each Technique

On the other hand, SNMP is not able to provide the same level of detail as Vicinity Sniffing, but it is guaranteed to report statistics from every packet that passes through that segment of the network. As shown above, the number of packets that are included in the reported statistics differs between Vicinity Sniffing and SNMP, but the graphs of the SNMP data have clear gaps whereas the Vicinity Sniffing data does not have time gaps. Interestingly enough, the most accurate measure of traffic comes from the less-detailed source. This is also illustrated in the following discussion.

Figure 23 displays the average throughput that was achieved by the client in Experiment 1. The graph is based on the data that was gathered with both SNMP and Vicinity Sniffing. In order to directly compare the two techniques, the Vicinity Sniffing data was averaged to reflect the same time intervals as the data acquired from SNMP. As indicated, the measured throughput differs slightly. This difference is due to the accuracy of the data gathering techniques. While Vicinity Sniffing is able to show more details, it is evident from the graph that some of the traffic was not captured for Experiment 1.

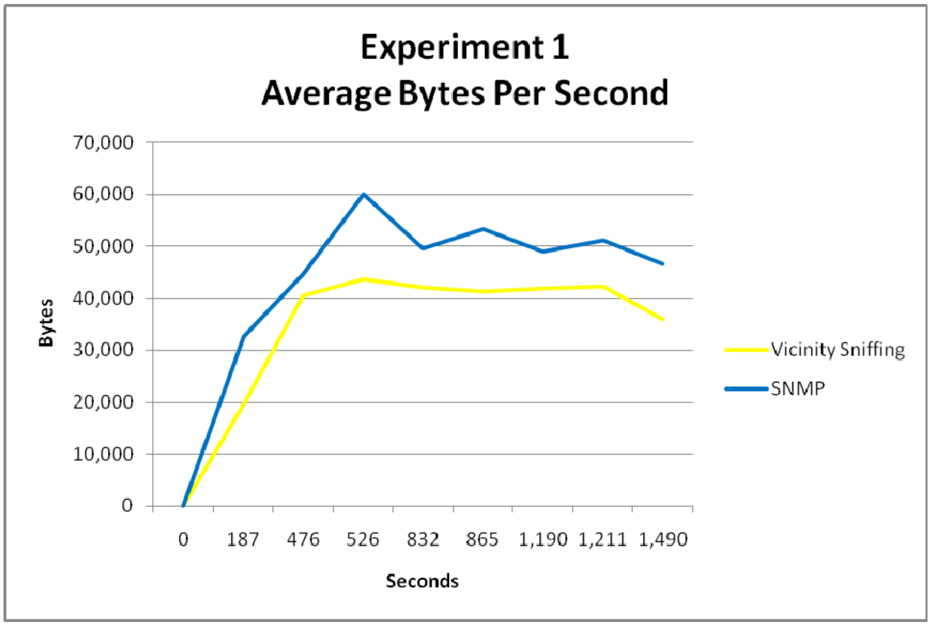


Figure 23: Experiment 1 – Average Bytes per Second

Figure 24 shows the average bitrates that were achieved for Experiment 3 for each measurement technique. Experiment 3 was conducted with the sniffers in a more desirable location than Experiment 2, which explains why the two curves in Figure 24 are relatively consistent.

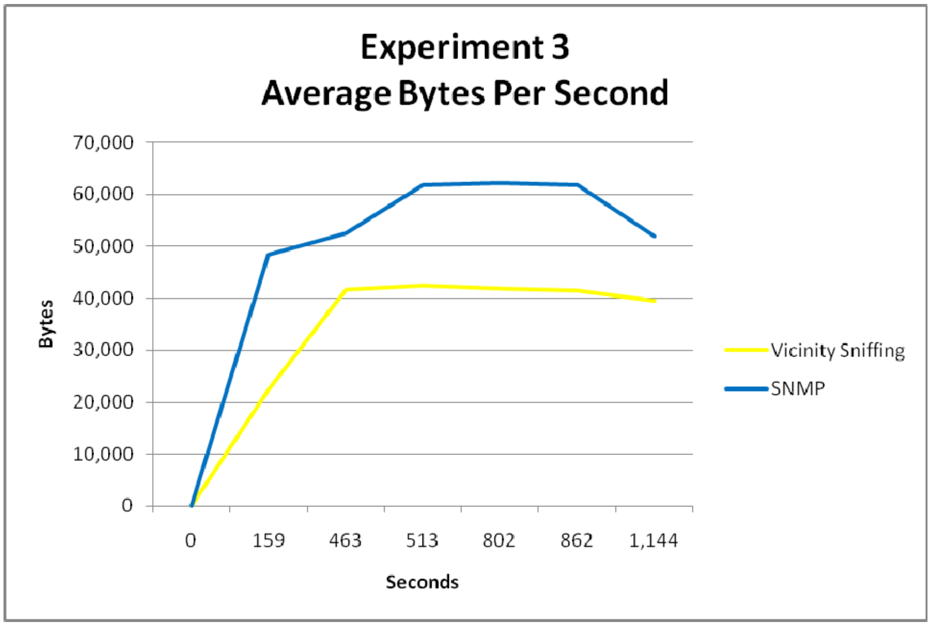


Figure 24: Experiment 3 – Average Bytes per Second

Table 9 shows another comparison of the two techniques. Here, we analyzed the results from the experiments and directly compare the number of RTS frames. This table highlights the difference in the measurement accuracy of the two techniques. We can see that there is a discrepancy in the total counts, and this can be explained by the number of packets and frames that were actually captured and reported. We see that the SNMP data reports about 54% more captured RTS frames than Vicinity Sniffing, but this can be attributed to the placement of the sniffers not being optimal, the configuration problems, and SNMP reporting on all frames.

Metric	*Vicinity Sniffing*	*SNMP*
Total RTS Frames	12,396	19,163

Table 9: Comparison of RTS Frames Captured by Technique

The network topology also greatly affected the data acquisition process. The fact that the iTiger network was engineered to use the Cisco Wireless Controller and the Lightweight Access Points allowed us to take advantage of the functionality. We were able to issue SNMP queries to one network device and retrieve all of the required information instead of having to query each device by itself. We further utilized the network setup and were able to use one general query instead of several individual queries.

These two reductions in SNMP queries were crucial in the success of this study. Our implementation actually had a negligible effect on the network performance. Due to the nature of how SNMP acquires data, the wireless controllers automatically track the statistics and update their own internal counters for the data that we were interested in analyzing. Because of this, the use of our SNMP-based data retrieval system may be considered as nonintrusive as Vicinity Sniffing.

In summary, we have observed that our results obtained from Vicinity Sniffing are comparable to the results that we acquired via SNMP. We have also found (through the analysis of the additional set of experiments) that our results are consistent with those that other researchers have obtained from their experiments. Additionally, we were able to directly compare the bitrates that were measured by both techniques as well as the total number of frames and RTS frames that were captured.

A tradeoff should be noted when using SNMP; some of the data that is returned may be aggregated between query intervals. Increasing the frequency of the SNMP queries beyond the device's update interval may generate traffic that consumes the available network resources, while at the same time returning the same data. Decreasing the frequency of the SNMP queries may lead to results that are not detailed enough to support the needs of the researcher(s).

On the other hand, analysis using Vicinity Sniffing can be quite cumbersome, and the number of sniffers required to accurately measure the network may be large. While Vicinity Sniffing can observe and capture details that are received by access points, obtaining the information can be difficult, and may even not be feasible in a real-time manner.

CONCLUSIONS

The use of SNMP as a tool for monitoring IEEE 802.11 Wireless networks has been presented. There exists a great deal of opposition and negative thoughts when it comes to using SNMP as a tool, or even using it as a basis for a tool. The main arguments against SNMP include the fact that data is reported as either aggregate statistics or as instantaneous values. While this is true, the available information is still very useful in its own right. Other arguments against SNMP include the claim that it cannot report statistics on a per-device level. This is not correct. As we have shown above, the SNMP MIB AIRESPACE-WIRELESS-MIB includes tables that are specifically dedicated to individual access points as well as individual clients.

In an effort to demonstrate that state-of-the-art wireless equipment can effectively produce SNMP statistics on a per-device level, we conducted several experiments and utilized two very different techniques for data acquisition. We used the common, and heavily utilized, methods of wireless monitoring, more specifically, we used the techniques of Vicinity Sniffing. We also used our SNMP-based data gathering methodology. We have shown that our Vicinity Sniffing results are generally consistent with those that were obtained from previous studies. We were able to correlate the throughput that was calculated from the two techniques and we separated the SNMP data into the individual experiments. We also compared the number of frames that each technique was able to capture, and in addition, we were able to provide statistical evidence to show that in some instances SNMP is capable of more accurately capturing network traffic than Vicinity Sniffing.

Throughout our studies, we have examined two different techniques for acquiring network statistics from IEEE 802.11 Wireless Networks. Each methodology was presented individually in an unbiased manner. Both of the techniques are accompanied by their own set of advantages and disadvantages. The use of one technique over another is entirely dependent upon the specific situation at hand, but both techniques are able to produce relatively accurate results. Contrary to previous claims, it is possible to track the behavior of individual users with either technique.

Our results and analysis serve as a proof-of-concept that SNMP-based tools are more accurate and more useful than previously thought, especially when coupled with current wireless equipment. We have also shown that more statistics are available from SNMP, and that it is capable of producing detailed statistics at the device-level. We have shown that the data obtained from using such tools is as reliable and accurate, and in some instances, may be even more accurate than the widely accepted methods of wireless monitoring such as Vicinity Sniffing.

We have shown that the latest generation of 802.11 wireless equipment is capable of providing data that was once thought to be obtained only from Wireless Monitoring. Being able to compare even a small number of results obtained from Vicinity Sniffing and SNMP is a significant contribution as it provides a valuable confirmation of our methods. Certain types of information, however, still require Vicinity Sniffing, such as individual MAC frame types. However, we expect future 802.11 standards to continue this blending of an SNMP and Wireless Monitoring. An emerging standard being developed by the IEEE and the Wi-Fi Alliance called 802.11v exemplifies this trend [24]. The proposed standard (expected completion is in mid-2010) creates an interface that allows Wi-Fi networks to be managed down to the client device. The enhancement will allow central resource management to obtain more detailed client performance data allowing the system to override client-based roaming decisions and to support location-aware applications.

To provide more validity and reliability in our studies we conducted an additional set of experiments. The details of which may be found in Appendix B. Overall, the data from this new set of experiments is consistent with the results that were found in previous research. This information provides a great deal of validity to both our previous results and our methodology for in acquiring data.

Through the analysis of our results, we are able to make a recommendation for the methodology that should be used by iTiger. We found that using shell scripts and Perl scripts to capture, filter, and store the SNMP data proved very effective. We also found that real-time monitoring is possible using a freely available tool called MRTG, and that post game processing of the SNMP data can be performed by many different tools including Microsoft Excel and Matlab. We found that our methodology supports post game processing by making the SNMP data available in CSV formatted data files. Our study contributes to the research community by providing a possible methodology for real-time network monitoring (via MRTG) and post game data processing. Our methodology supports the needs of researchers by making available the SNMP data, predominantly the per-client data.

REFERENCES

[1] J. Yeo, M. Youssef, and A. Agrawala. “A Framework for Wireless LAN Monitoring and its Applications.” http://portal.acm.org/citation.cfm?id=1023646.1023660

[2] J. Yeo, M. Youssef, and A. Agrawala. “Characterizing the IEEE 802.11 Traffic: Wireless Side.” http://www.cs.umd.edu/~moustafa/papers/CS-TR-4570.pdf

[3] J. Yeo, M. Youssef, T. Henderson, A. Agrawala. “An Accurate Technique for Measuring the Wireless Side of Wireless Networks.” http://portal.acm.org/citation.cfm?id=1072433

[4] M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, J. Zahorjan. “Measurement-based Characterization of 802.11 in a Hotspot Setting.” http://portal.acm.org/citation.cfm?id=1080150

[5] J. Kantorovitch, P. Mahonen. “Case studies and experiments of SNMP in wireless networks.” http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1045776&isnumber=22406

[6] K. Papagiannaki, R. Cruz, C. Diot. "Network Performance Monitoring at Small Time Scales." http://www.imconf.net/imc-2003/papers/paper1.pdf

[7] R. Raghavendra, E. Belding, K. Papagiannaki, K. Almeroth. "Understanding Handoffs in Large IEEE 802.11 Wireless Networks." http://www.imconf.net/imc-2007/papers/imc192.pdf

[8] J. Yeo, S. Banerjee, A. Agrawala. “Measuring traffic on the wireless medium: experience and pitfalls.” http://www.cs.umd.edu/Library/TRs/CS-TR-4421/CS-TR-4421.pdf

[9] Jardosh, K. Ramachandran, K. Almeroth, E. Belding-Royer. “Understanding Congestion in IEEE 802.11 Wireless Networks.” http://www.usenix.org/events/imc05/tech/full_papers/jardosh/jardosh_new.pdf

[10] M. Balazinska, P. Castro. “Characterizing Mobility and Network Usage in a Corporate Wireless Local-Area Network.” http://portal.acm.org/citation.cfm?id=1066127

[11] Balachandran, G. Voelker, P. Bahl, P. Rangan. “Characterizing User Behavior and Network Performance in a Public Wireless LAN.” http://sysnet.ucsd.edu/pawn/papers/wireless_sig.pdf

[12] T. Henderson, D. Kotz, I. Abyzov. "The Changing Usage of a Mature Campus-wide Wireless Network." http://portal.acm.org/citation.cfm?id=1023720.1023739

[13] Jardosh, K. Ramachandran, K. Almeroth, E. Belding-Royer. "Understanding Link-Layer Behavior in Highly Congested IEEE 802.11 Wireless Networks." http://portal.acm.org/citation.cfm?id=1080151

[14] Y. Cheng, M. Afanasyev, P. Verkaik, P. Benko, J. Chiang, A. Snoeren, S. Savage, G. Voelker. "Automating Cross-Layer Diagnosis of Enterprise Wireless Networks." http://www-cse.ucsd.edu/~savage/papers/Sigcomm07.pdf

[15] IEEE 802.11. (2009, March 8). In Wikipedia, The Free Encyclopedia. Retrieved 13:41, March 8, 2009, from http://en.wikipedia.org/wiki/802.11

[16] Internet Protocol Suite. (2009, March 8). In Wikipedia, The Free Encyclopedia. Retrieved 13:45, March 8, 2009, from http://en.wikipedia.org/wiki/TCPIP

[17] Carrier sense multiple access with collision avoidance. (2009, March 8). In Wikipedia, The Free Encyclopedia. Retrieved 13:55, March 8, 2009, from http://en.wikipedia.org/wiki/CSMA_CA

[18] Mauro, Douglas R., Schmidt, Kevin J. *Essential SNMP*. O'Reilly Media, Inc. Sebastopol, 2005.

[19] IEEE802dot11-MIB. Retrieved 15:45, March 16, 2009 from http://www.oidview.com/mibs/0/IEEE802dot11-MIB.html

[20] AIRESPACE-WIRELESS-MIB. Retrieved 15:45, March 16, 2009 from http://www.oidview.com/mibs/14179/AIRESPACE-WIRELESS-MIB.html

[21] A. Mahanti, M. Arlitt, C. Williamson. "Assessing the Completeness of Wireless-side Tracing Mechanisms." http://pages.cpsc.ucalgary.ca/~carey/papers/2007/aniket-wowmom2007.pdf

[22] C. Mathias. "Cisco Spectrum Expert sets pace for affordable spectrum analysis tools". Network World. 03/23/2009. http://www.networkworld.com/reviews/2009/032309-wlan-test-cisco.html

[23] "Microsoft Excel: Some other numbers …" Retrieved 21:34, April 3, 2009 from http://blogs.msdn.com/excel/archive/2005/09/26/474258.aspx

[24] J. Wexler. "Introducing 802.11v - a hope for Wi-Fi management". Network World. 01/21/2005. http://www.techworld.com/networking/features/index.cfm?featureid=1135

[25] “Cisco Systems Completes Acquisition of Airespace”. Cisco Newsroom, March 24, 2005. http://newsroom.cisco.com/dlls/2005/corp_032405.html?CMP=ILC-001

APPENDICES

Appendix A:

The SNMP-Based Wireless Monitoring Methodology

Another goal of our study was to produce a methodology that would become available to future researchers and to support the research efforts of Clemson University's iTiger project. The methods did not exist while we were conducting our evaluation, so they had to be created on the fly.

For instance, initially we wanted to be able to automate the process of merging the network captures that were acquired from the Vicinity Sniffing techniques with the data that was gathered from SNMP. This appeared to be a trivial problem, but what we encountered was unexpected. The information gathered from both techniques had different reference points that could not be easily compared. One example is the reference of time. In the wireless monitoring experiments, each device started and stopped the capture process and recorded its own timestamp. The SNMP data turned out to not contain enough information to discern a specific time for a specific event. As previously noted, SNMP data is based on either aggregate statistics or instantaneous values. The only information that we were able to obtain to serve as a time reference was the timestamp of when the information was retrieved by our tool. This meant that we could not compare the two sources and correlate an exact point in time. Instead, we were able to examine the results and analysis from each source and deduce a correlation of events.

Another requirement that had to be changed involved the visual representation of the data. Again, we wanted this to be automated. We found that in a real-life setting, the exact set of clients would be unknown, and without a static list, the graphing tools could not maintain the required statistics. Because of that, we chose to automate the data processing and output the results in a format that could be read and manipulated by other programs. We decided to store the SNMP data in CSV files and import them into Microsoft Excel. From there, we created the graphs and charts that appear above manually.

Despite the challenges, we did succeed in defining a methodology. The basic operation is simple, clean, and concise. The operation is composed of shell scripts, a Perl script, and access to a SNMP-enabled device. The shell scripts drive the entire process. They initiate the SNMP query, store the returned data in a text file, and then call the Perl script that parses and filters the data, then formats and saves it to a set of CSV files. After all of the querying is completed, another shell script combines all of the CSV files that describe the access points into one larger file, and all of the CSV files that describe the clients into another CSV file. Once the CSV files are created and merged, they can be imported into various graphing tools.

Figure 25 and Figure 26 show the hierarchical decomposition of the SNMP OIDs that we were interested in observing. One thing to note is that the MAC address of the wireless client is used to index into the SNMP table *bsnMobileStationTable*, but it is stored in base-10. This means that the Hexadecimal MAC address is converted into decimal notation and instead of using colons ":" as the separator it uses periods ".". The images below uses the notation "(Dec)" to refer to the decimal version of the MAC address and "(Hex)" to refer to the Hexadecimal version. Also displayed is either the numeric version of the OID, or the manner with which that information was obtained.

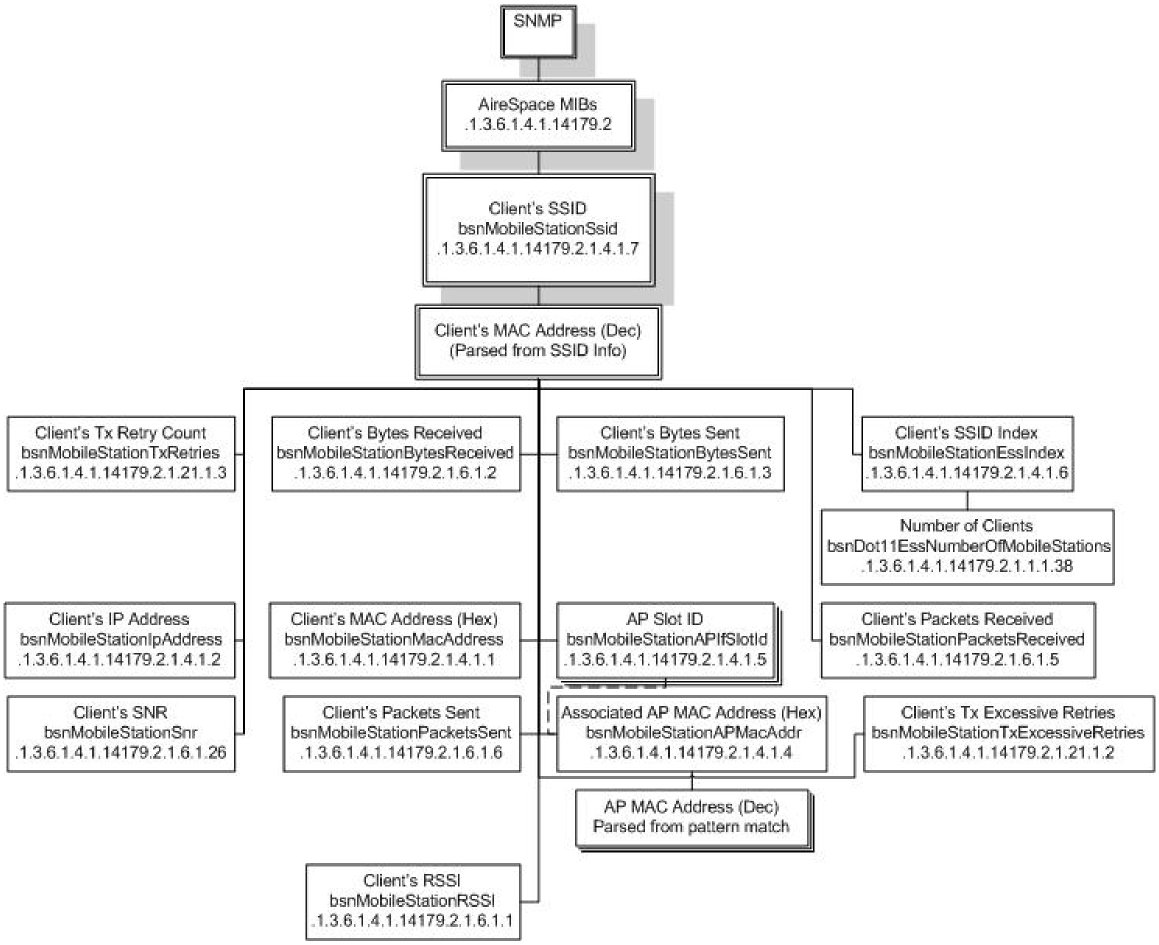


Figure 25: Hierarchical Decomposition of the Captured SNMP OIDs

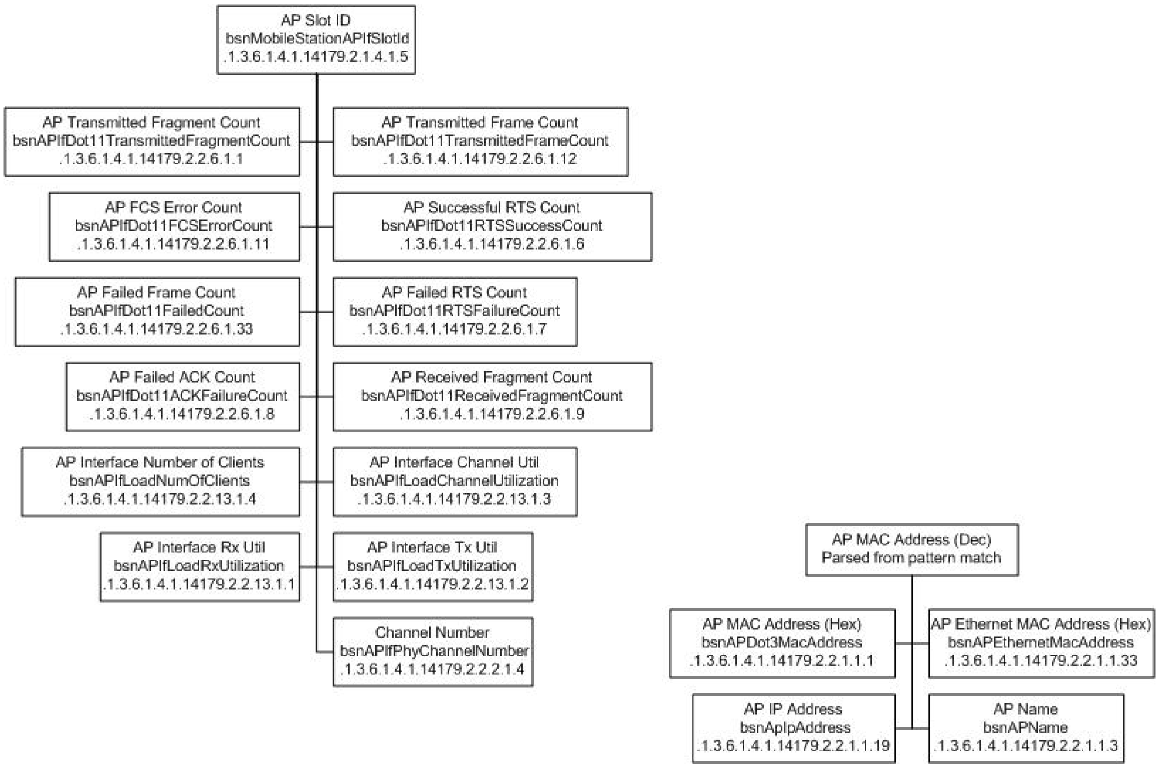


Figure 26: Further Details of the AP Slot ID Tree and the AP MAC Address Tree

In addition to creating the network monitoring and measurement methodology, we also established the use of MRTG[13] on the iTiger network. MRTG is a graphing tool that utilizes SNMP to retrieve network statistics and creates graphs in real-time or near to real-time. It is a piece of Open Source software that is written in Perl and works on a wide array of operating systems including Windows, Unix/Linux, and Netware systems. Although it can be extended, the primary function of MRTG is to monitor SNMP capable devices and graph the amount of network traffic that passes through each interface.

[13] "The Multi Router Traffic Grapher" (http://oss.oetiker.ch/mrtg/)

The use of MRTG on the iTiger network allows researchers to monitor the network traffic in real-time during a football game. Alternatively, if they so desire, the network statistics may be extracted after the game has completed. This shows that SNMP is being used in more research environments, and it directly supports the idea that future developments of network equipment will continue to advance the usage of this very versatile tool.

Appendix B:

Results and Analysis from Additional Experiments

In an effort to provide more validity to the results and conclusions that were presented above, we chose to conduct an additional set of experiments. Just as before, we used the Vicinity Sniffing techniques in conjunction with gathering data from SNMP. Neither the hardware nor any of the software was changed from the original set of experiments, we also used the same network setup as the previous experiments (see Figure 2), but we focused on a more controlled environment for the second set. One of the controlled variables was the wireless channel that the devices used to connect. In our previous experiments, we used both channels one and eleven. In this new set of experiments, all of the devices were configured to use a single channel, six.

Table 10 describes the locations of the wireless monitoring sniffers and the clients for each experiment. Figure 27 shows a map of Clemson University's West Zone Club augmented with the location identifiers. We show that the locations of the sniffers were not changed during the second set of experiments, and only the clients were moved. From the original data, we found that sniffers at locations A and B captured the largest number of frames, so we decided to use those two locations for the new set of experiments. In addition, five experiments were conducted instead of the previous four to obtain a larger set of data. Just as with the previous experiments, all of the wireless clients downloaded the same previously recorded video of a halftime show from one of Clemson's football games.

Experiment Number	*Sniffer Locations*	*Client Locations*
1	A, B	1, 2
2	A, B	1, 2, 3
3	A, B	1, 2, 4
4	A, B	2, 3, 4
5	A, B	1, 3, 4

Table 10: Additional Experiment Configurations

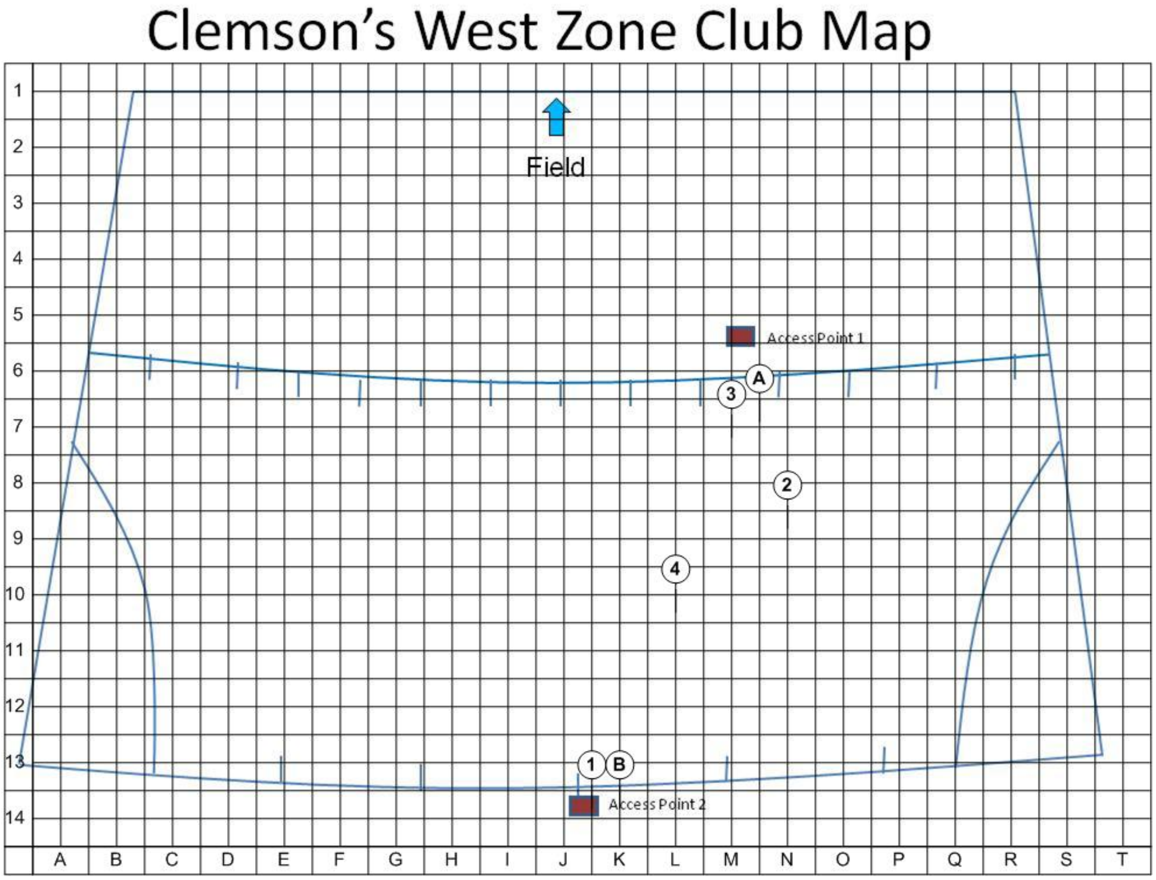


Figure 27: Map of Additional Experiment Locations

Results and Analysis

Just as in the previous set of experiments, we used both methods of network measuring to obtain our data. We present each method individually, and then present the comparison of bitrates that were observed from each technique.

Vicinity Sniffing

As previously stated, five new experiments were conducted. Table 11 presents a summary of the experiments in the same format as Table 4. Listed are the total number of packets that were captured, average bitrates, and the average packet sizes. The packets that are included in these statistics are from the combination of the network captures from both sniffers for each experiment. Just as before, the experiment that is labeled “All” is the result of merging the individual capture files into one large capture file. The statistics were then extracted from the single file to present a general overview of the experiments.

Experiment	***Number of Packets Recorded***[14]	***Data Rate KB/sec.***	***Average Packet Size (Bytes)***
1	213,411	27.1	150.57
2	231,667	30.5	161.02
3	226,046	25.1	143.49
4	185,399	26.5	158.56
5	215,269	23.5	143.18
All[15]	1,071,792	26.4	151.24

Table 11: Summary of Additional Vicinity Sniffing Experiments

Figure 28 and Figure 29 shows an overview of the data rate changes over time for all of the experiments in frames per second and bytes per second, respectively. It is clear to see that the overall throughput remained relatively constant throughout the duration of the experiments. These results are consistent with the findings in [1, 2, 3, and 8].

[14] The number of packets captured is dependent upon placement of the sniffers in relation to the access points, location of the clients, and physical barriers that prevent the sniffer from observing the traffic.

[15] Not the sum of the statistics of the individual experiments; rather, it is the result of merging the capture files from each experiment from the additional set into one large capture file.

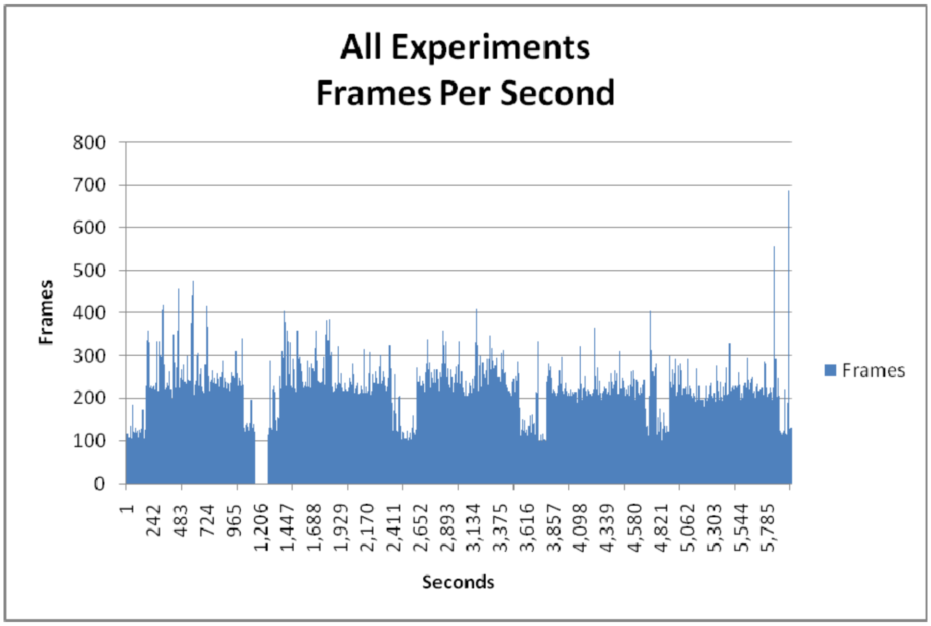


Figure 28: All Experiments – Frames per Second

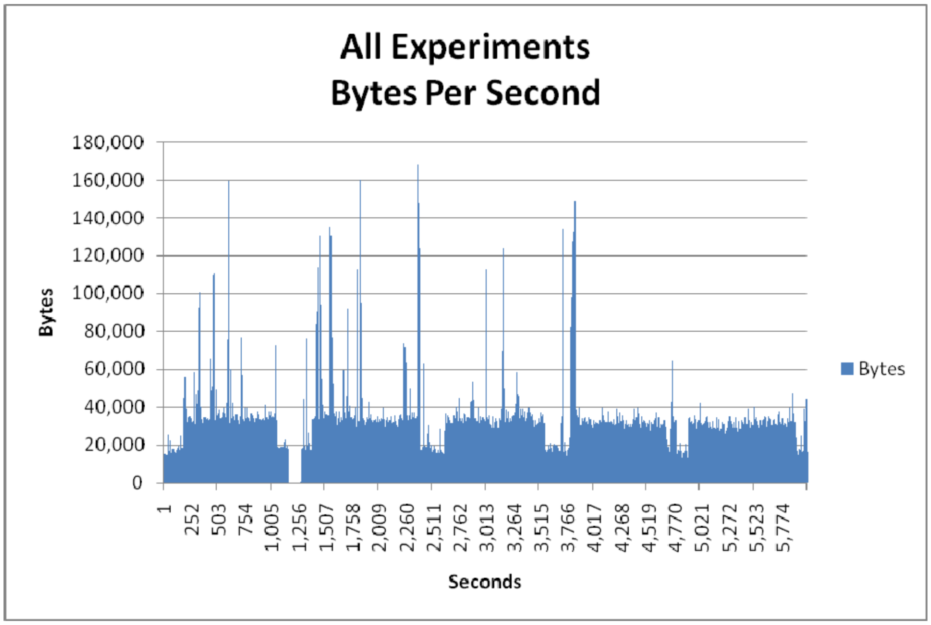


Figure 29: All Experiments – Bytes per Second

Similar to above, Table 12 presents the overall traffic characteristics of the frames, and these are visually displayed in Figure 30. Consistent with our previous findings, the largest percentage of observed frames is associated with IEEE 802.11 Beacon Frames. Also consistent with our original data is the fact that there is a significant difference in the number of observed RTS/CTS frames.

Frame Type	***Count***	***Percentage***
Association Request	562	0.0545%
Association Response	2,374	0.2300%
Reassociation Request	302	0.0293%
Reassociation Response	635	0.0615%
Probe Request	103,370	10.0166%
Probe Response	180,917	17.5309%
Beacon Frame	575,522	55.7681%
Authentication	2,939	0.2848%
Deauthentication	625	0.0606%
Power Save Poll	184	0.0178%
Request to Send	11,974	1.1603%
Clear to Send	38,748	3.7547%
Acknowledgement	113,840	11.0311%

Table 12: All Experiments – Frame Types

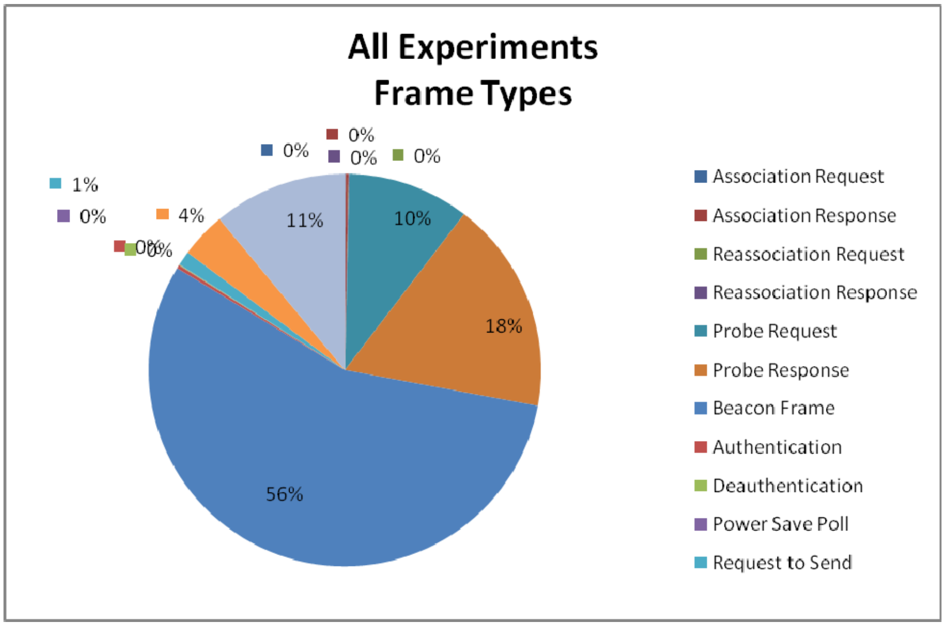


Figure 30: All Experiments – Frame Types

The results from the additional set of experiments are all very similar to each other and consistent with both our original results as well as the previously mentioned research. Because of this, only the frames-per-second bitrate graphs are shown here. The bytes-per-second bitrate graphs are discussed below where they may be compared to the SNMP-based bitrate graphs.

Figure 31, Figure 32, Figure 33, Figure 34, and Figure 35 display the bitrate that was achieved for each of the individual experiments. As shown in Table 13, the average number of frames that were observed each second is relatively consistent with each experiment. There are some minor rate differences, but the difference between the fastest experiment and the slowest experiment is about 25.39 frames per second. This shows that even though the locations of the clients changed for each experiment, the average speed did not significantly change. These findings are also consistent with the previous research that was conducted in [1, 2, 3, and 8] and provide additional validity to our results mentioned above.

Experiment Number	*Average Frames per Second*
1	183.98
2	193.70
3	179.12
4	170.87
5	168.31
All	178.54

Table 13: Average Frames per Second

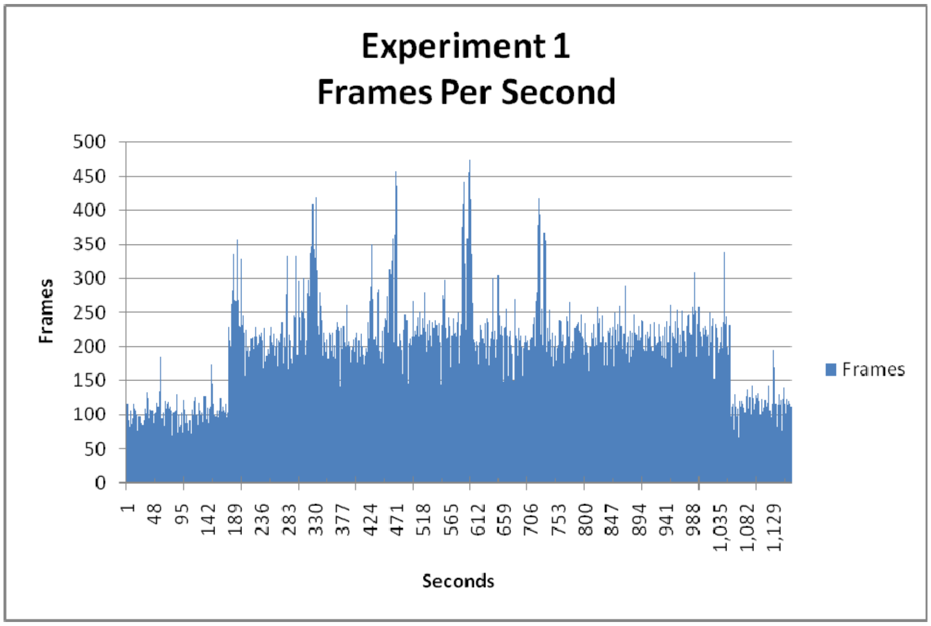


Figure 31: Experiment 1 – Frames per Second

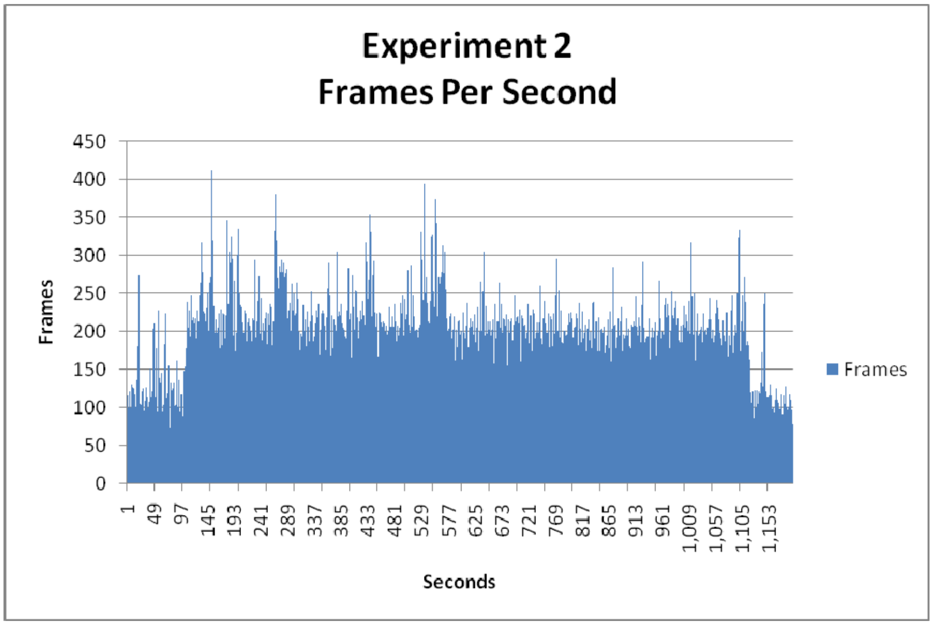


Figure 32: Experiment 2 – Frames per Second

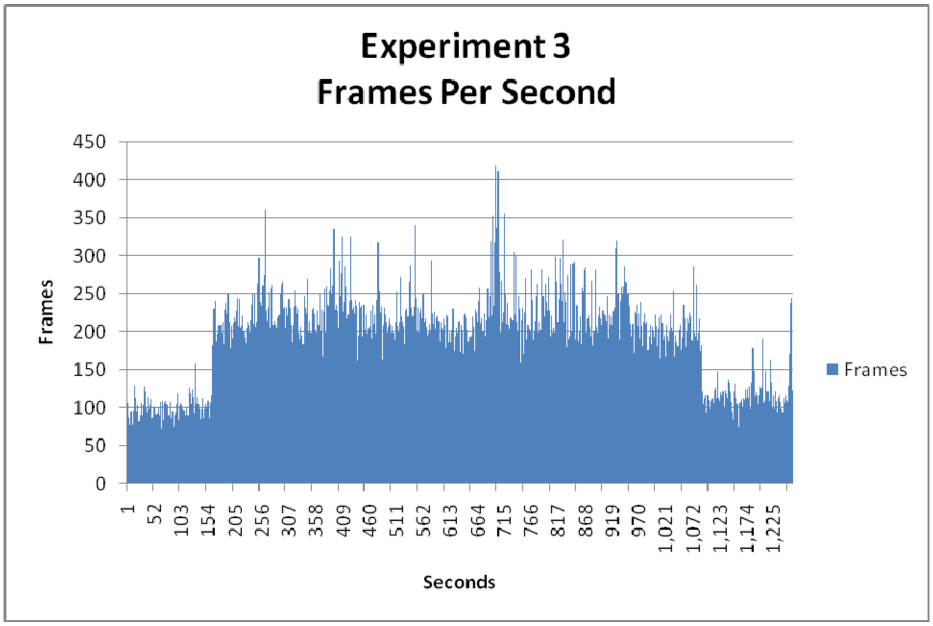


Figure 33: Experiment 3 – Frames per Second

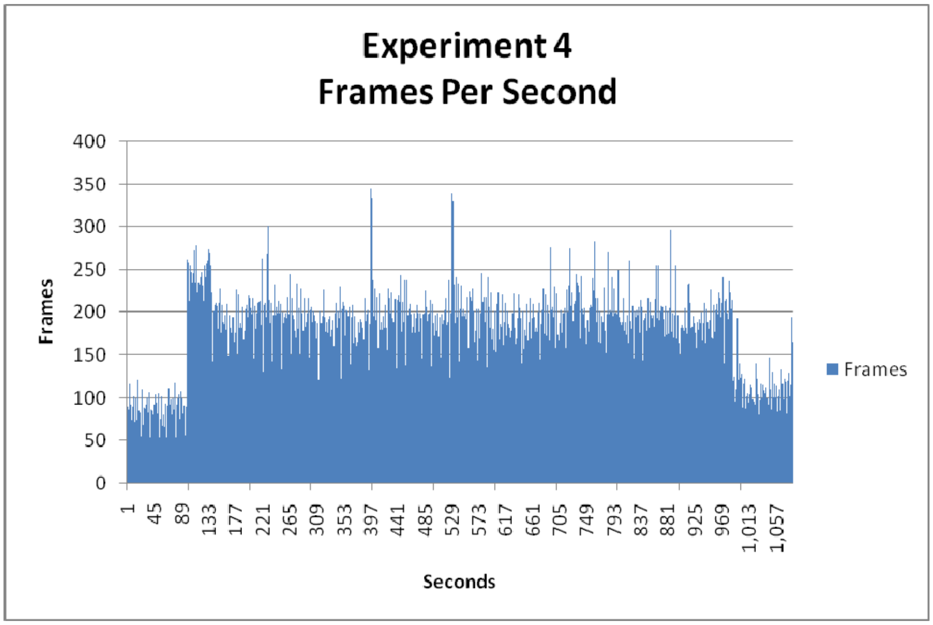


Figure 34: Experiment 4 – Frames per Second

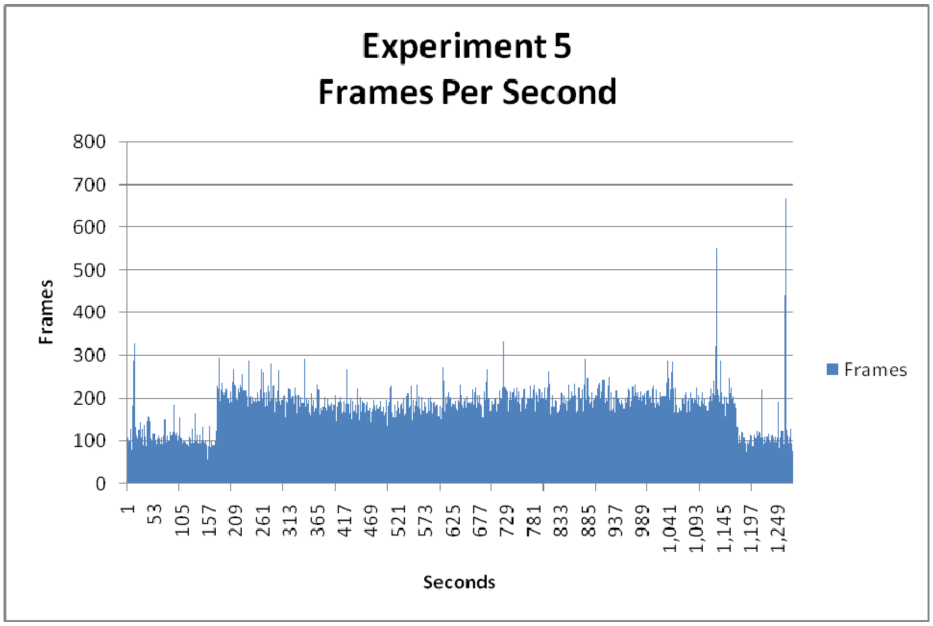


Figure 35: Experiment 5 – Frames per Second

SNMP

Similar to our previous methodology, while we were using Vicinity Sniffing to capture packets that were observed in the air, we also gathered SNMP data from the same computer as before. In addition to the Vicinity Sniffing data, the data that was acquired from the use of SNMP is also consistent with our previous findings.

Figure 36 shows the total number of packets that were sent and received by the clients, while Figure 37 shows the total number of bytes that were sent and received.

Consistent with our previous results, the number of packets that are sent by the clients follows a similar trend as the number of packets that are received. This is not the case when it comes to the total number of bytes. As shown below, the number of bytes that are sent by the clients is insignificant compared to the number of bytes that are received. Again, this difference is due to the size of the packets.

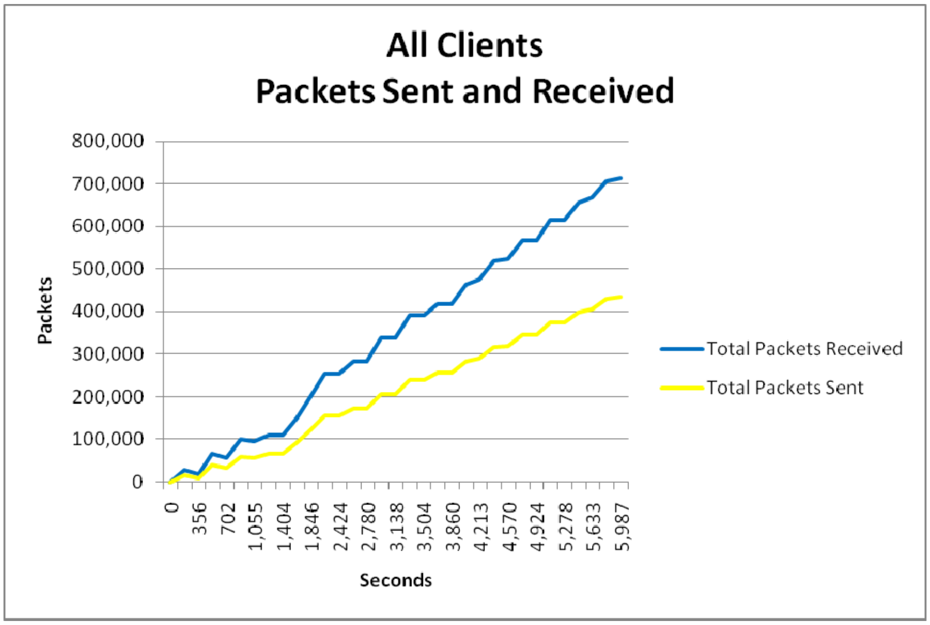


Figure 36: All Clients – Packets Sent and Received

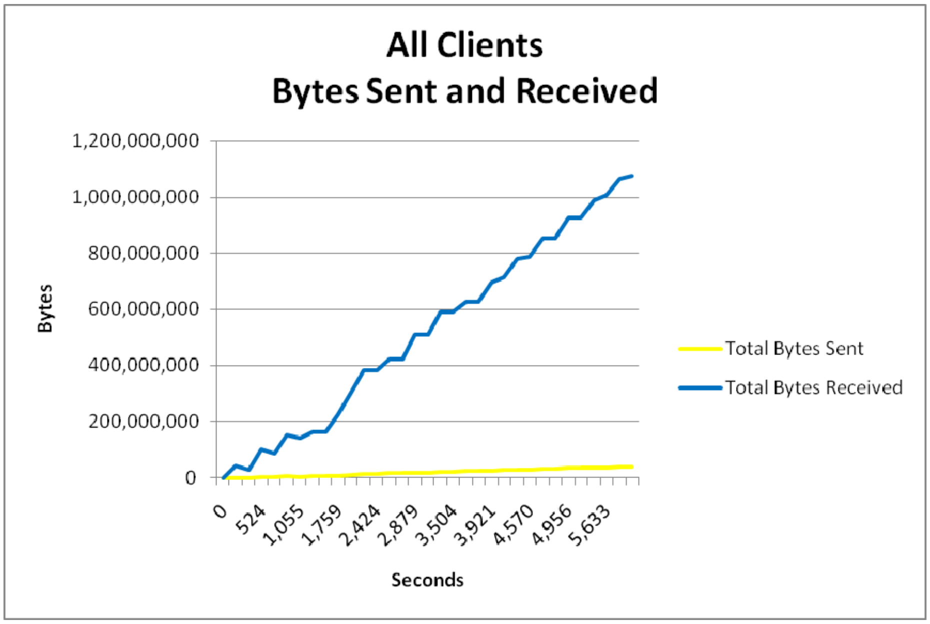


Figure 37: All Clients – Bytes Sent and Received

Below, the statistics from the AP viewpoint are displayed. The total number of failed/successful RTS frames is shown in Figure 38. Figure 39 shows the average transmission and channel utilization that was observed by the access points that were involved in the tests. From the graph, the time gaps between experiments can be seen when the utilization is at a low point, this is the most obvious with the channel utilization. The RTS frame statistics and the utilization information is also consistent with our previous findings, thus adding more validity to our results.

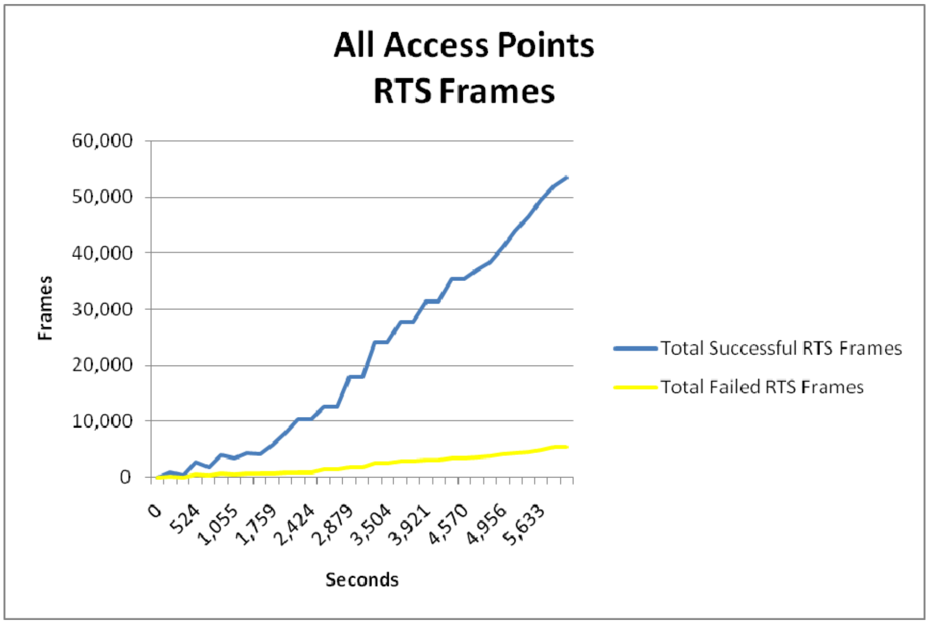


Figure 38: All Access Points – RTS Frames

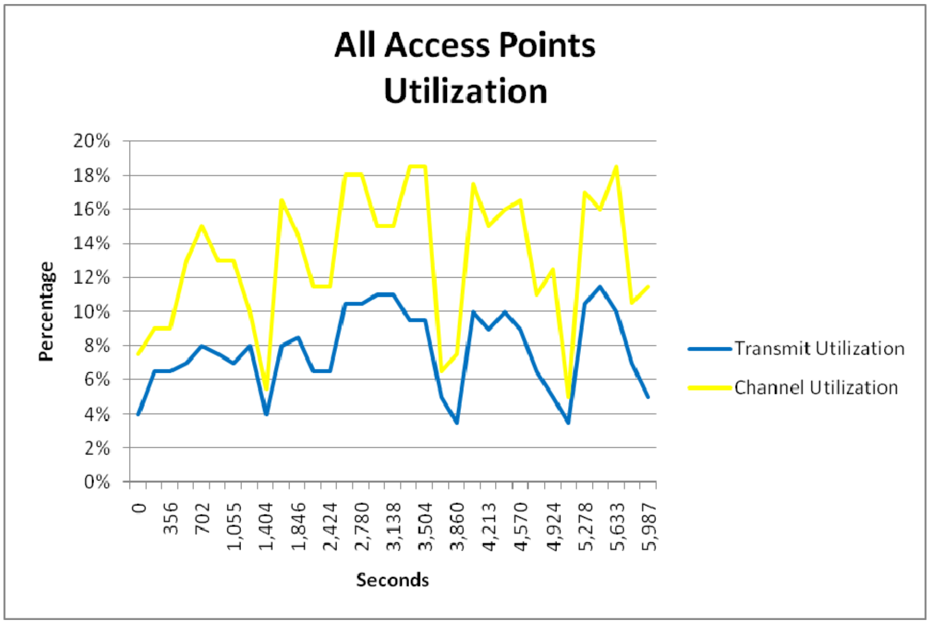


Figure 39: All Access Points – Average Utilization

The graphs below were generated from the SNMP data. Along with our previous results, these graphs disprove the claim that device-specific data cannot be acquired from SNMP. Figure 40 displays the number of successful RTS frames and the number of failed RTS frames per access point. From this information, it can be seen that both APs maintained a steady rate, but that one AP had a much higher percentage of successes than the other AP. Additionally, this behavior is the same that we observed in the previous set of experiments, as shown in Figure 22.

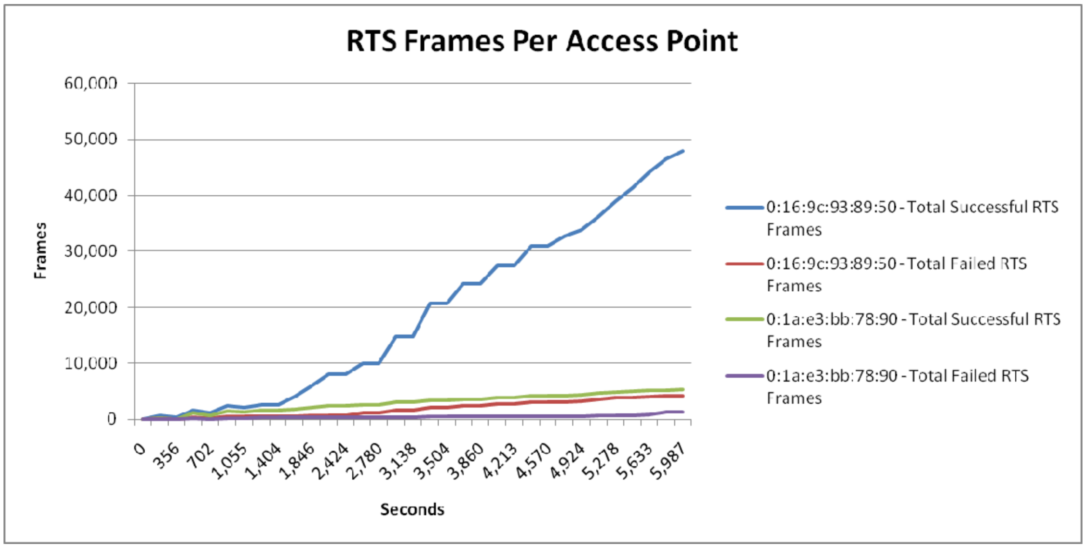


Figure 40: RTS Frames per Access Point

Figure 41, Figure 42, and Figure 43 all show statistics that are on a per-client basis. The average RSSI of each client can be seen in Figure 41. There is a greater level of detail than when the statistics from all of the clients are averaged together, something that is claimed to not be possible with SNMP.

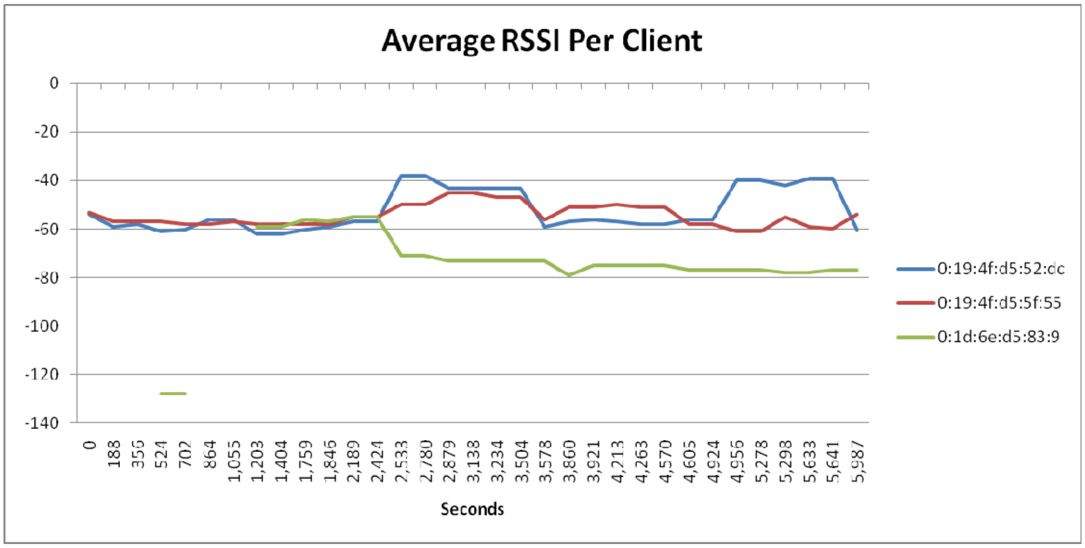


Figure 41: Average RSSI per Client

The number of packets sent and received per client is seen in Figure 42, and Figure 43 shows the number of bytes that were received per client. Similar to above, the amount of data that was transmitted by the clients is overshadowed by the amount that they received, and was thus omitted from the graph. Both of these figures are similar to those that were created from the data acquired from our original experiments.

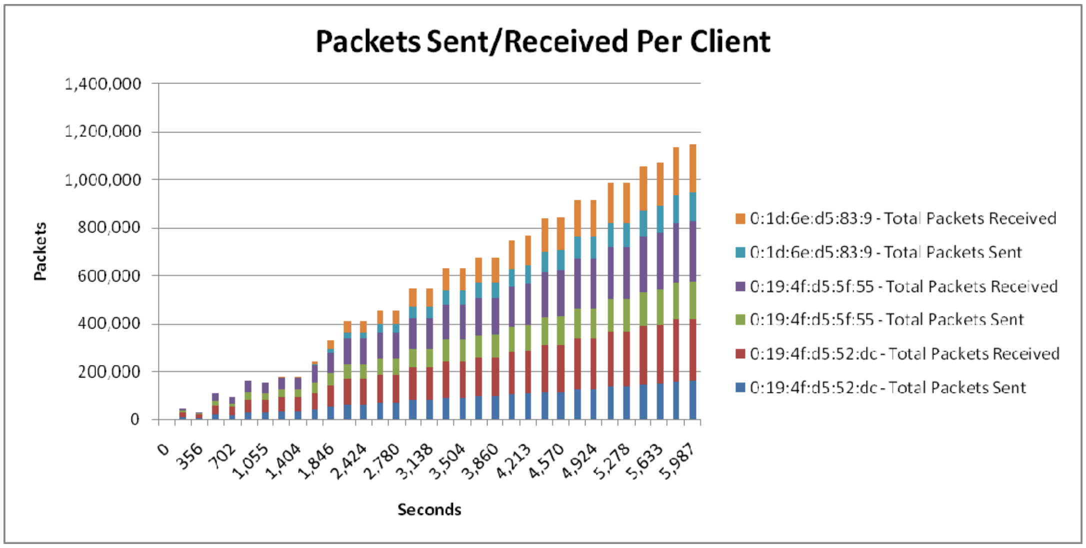


Figure 42: Packets Sent and Received per Client

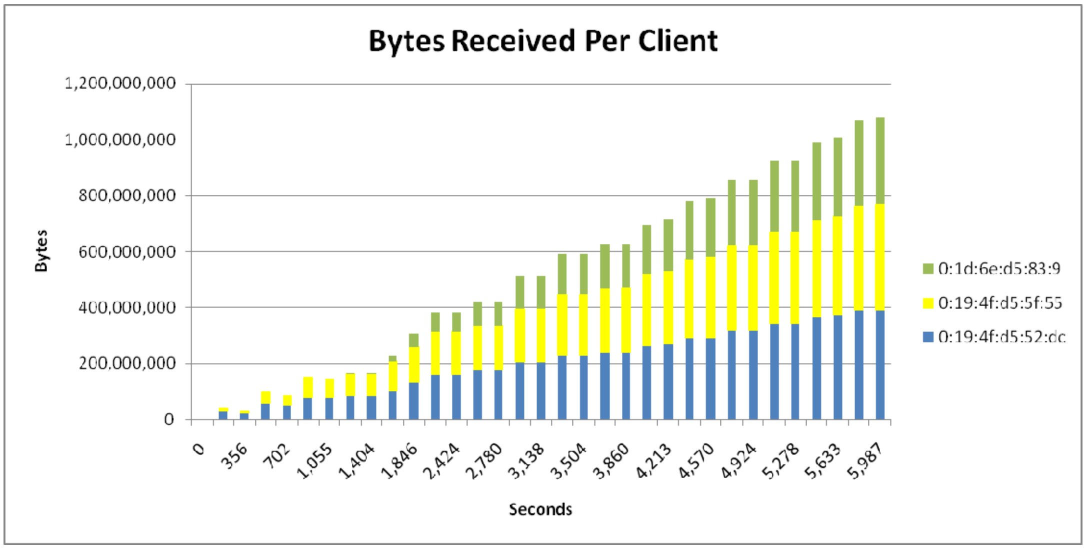


Figure 43: Bytes Received per Client

Comparison

Table 14 shows a comparison of the number of RTS frames as well as the total number of frames that were captured by the two different techniques. The difference between the two techniques is due to the physical characteristics of the data gathering methods, which were highlighted above. Overall, the techniques from Vicinity Sniffing captured 86% of the RTS frames that SNMP was able to capture, while at the same time capturing about 77% of the total number of frames that SNMP observed.

Metric	***Vicinity Sniffing***	***SNMP***
Total RTS Frames	50,722	58,936
Total Frames Captured	1,071,792	1,392,957

Table 14: Comparison of Frames Captured by Technique

The remaining graphs display the bitrates that were observed from both techniques, compared by experiment with the SNMP graphs on top and the Vicinity Sniffing graphs on the bottom. The SNMP-based graphs are shown with stacked values so that the total throughput may be seen as well as each direction. As shown, the data that was acquired from SNMP is consistent with the data that was recorded via Vicinity Sniffing. The SNMP-based bitrates are slightly higher than those of Vicinity Sniffing, and this can be explained by the fact that SNMP data encompasses all of the packets that cross the network and the Vicinity Sniffing data is only based on the packets that are captured.

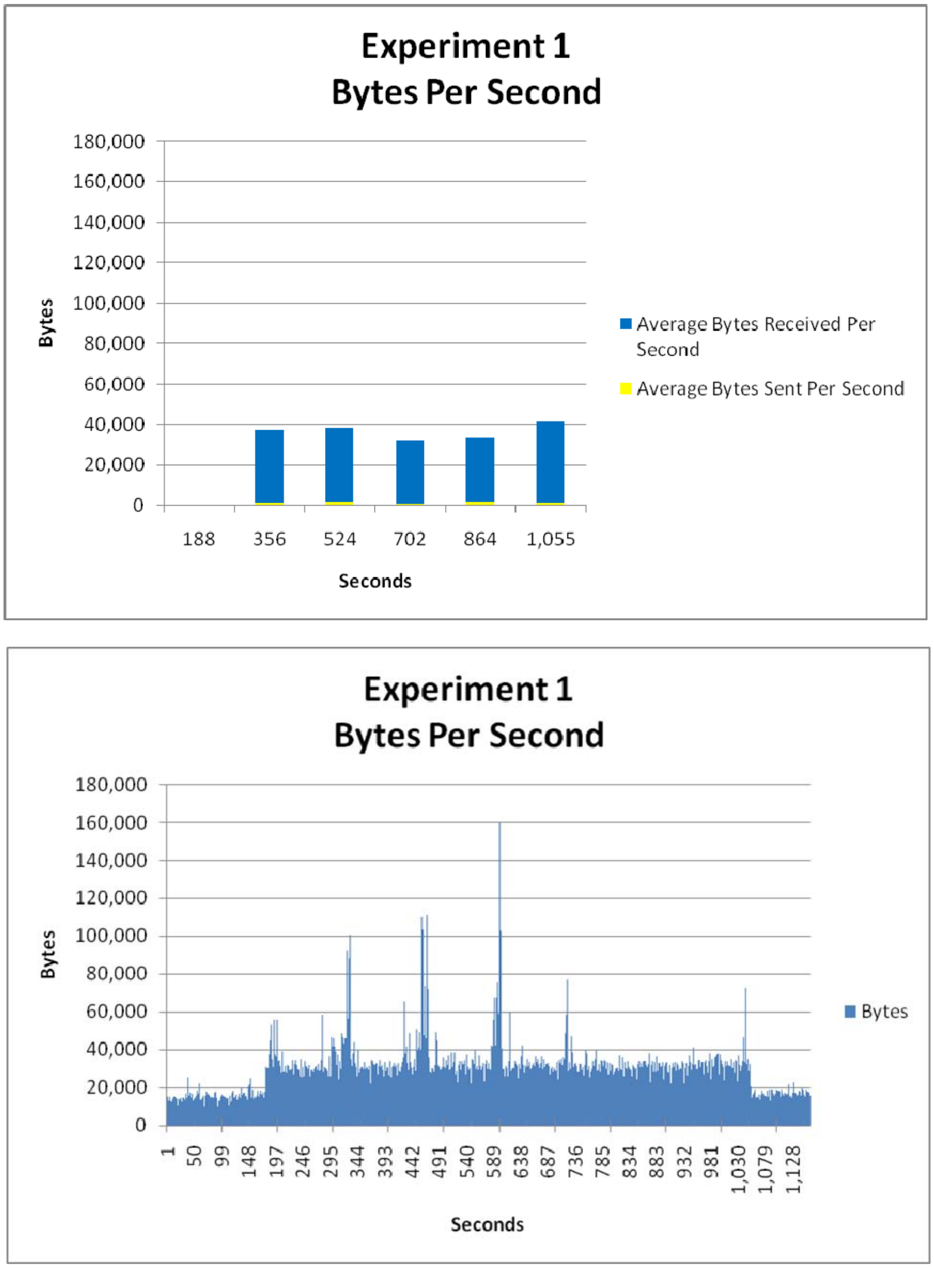


Figure 44: Experiment 1 – Comparison of Bitrates

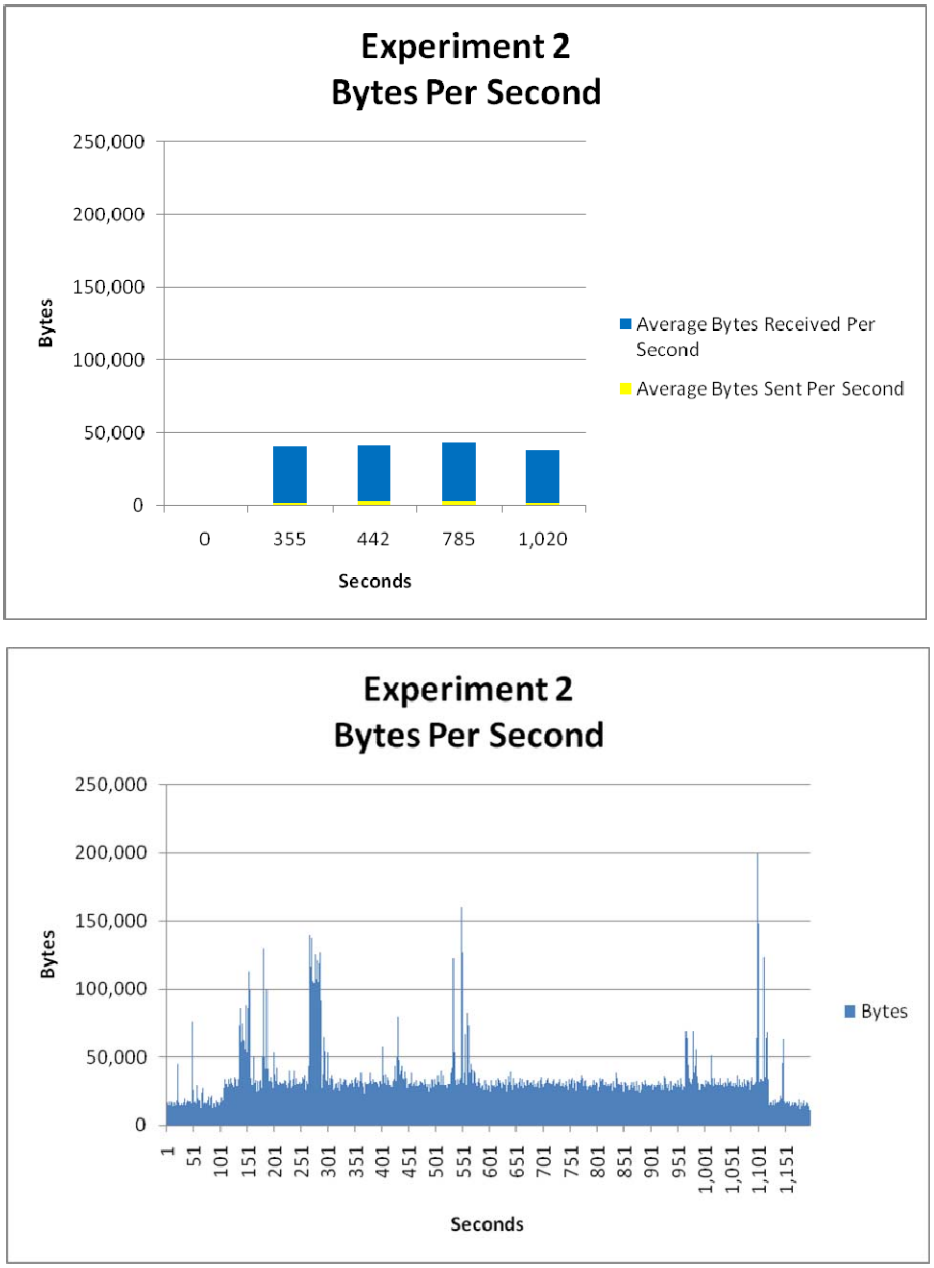


Figure 45: Experiment 2 – Comparison of Bitrates

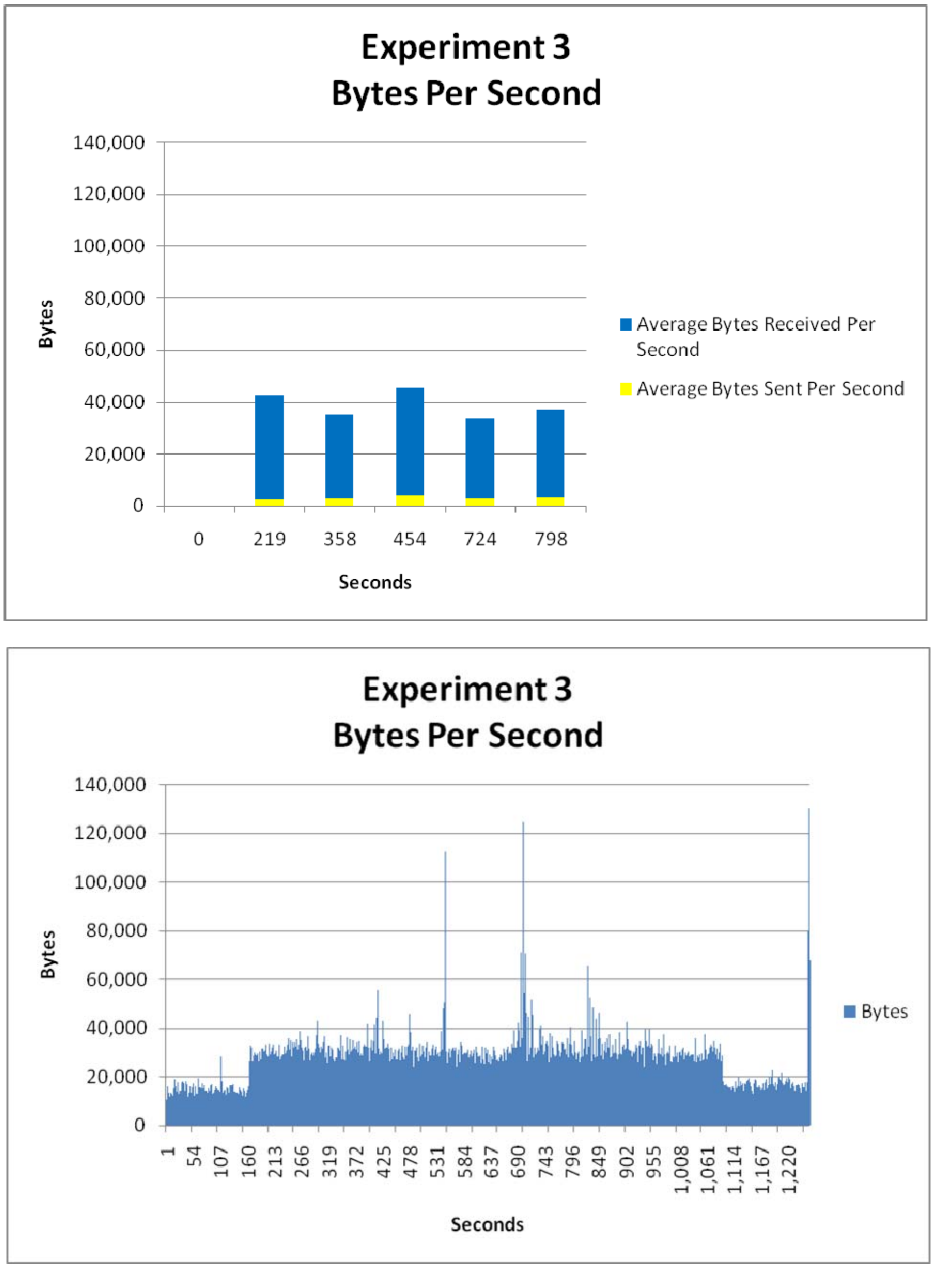


Figure 46: Experiment 3 – Comparison of Bitrates

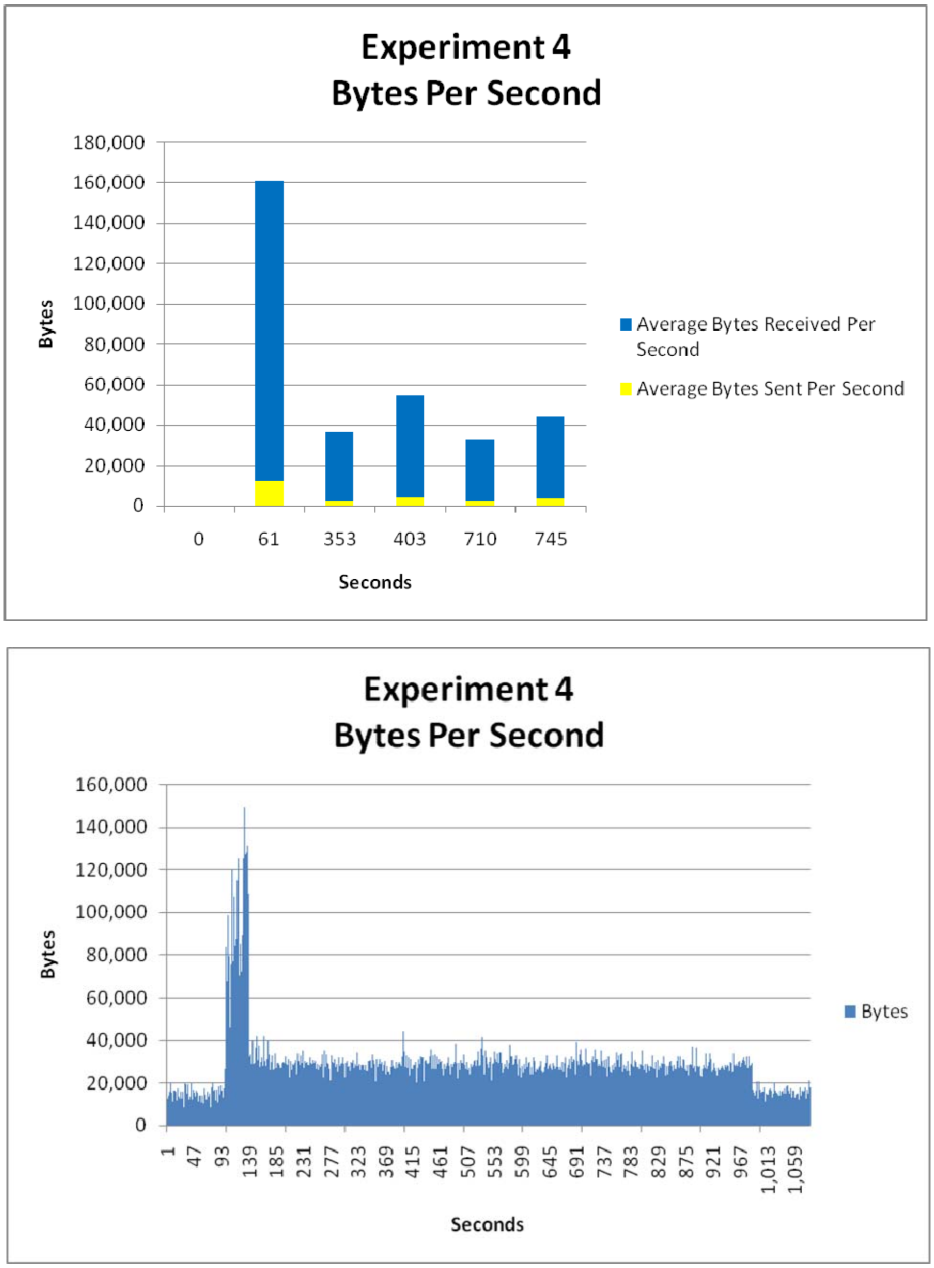
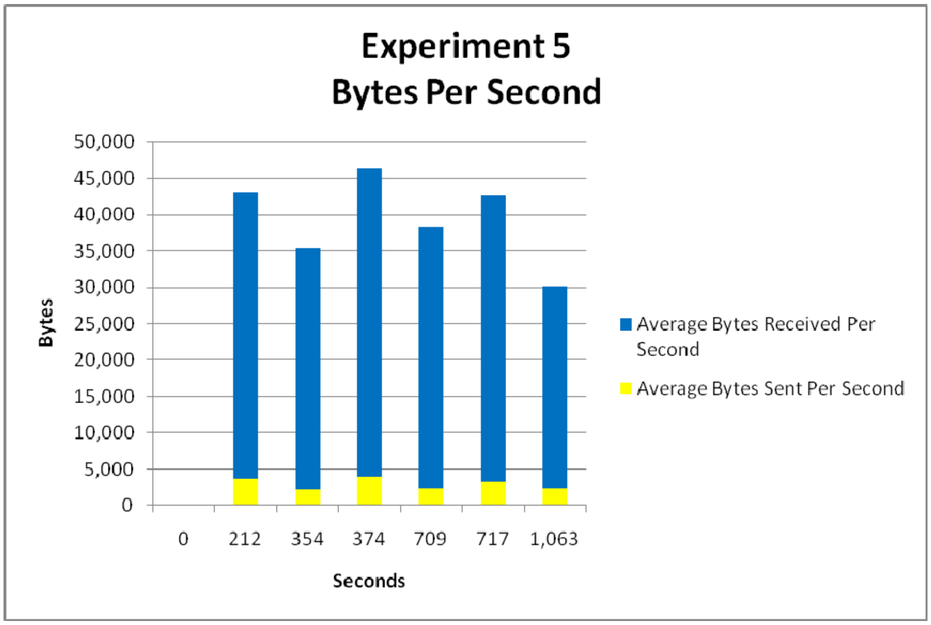


Figure 47: Experiment 4 – Comparison of Bitrates



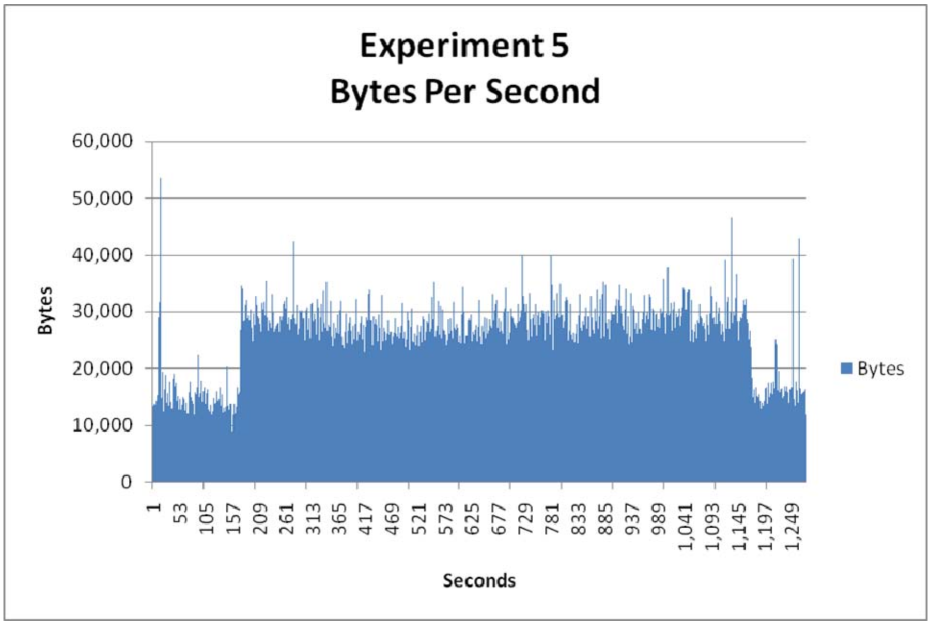


Figure 48: Experiment 5 – Comparison of Bitrates