



**PERFORMANCE ANALYSIS OF A SECURE IEEE 802.11B WIRELESS
NETWORK INCORPORATING PERSONAL DIGITAL ASSISTANTS**

THESIS

John Lee Camp, Captain, USAF

AFIT/GCS/ENG/02-10

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

Approved for public release; distribution unlimited

Report Documentation Page

Report Date Jun 02	Report Type Final	Dates Covered (from... to) -
Title and Subtitle Performance Analysis of a Secure IEEE 802.11B Wireless Network Incorporating Personal Digital Assistants	Contract Number	
	Grant Number	
	Program Element Number	
Author(s)	Project Number	
	Task Number	
	Work Unit Number	
Performing Organization Name(s) and Address(es) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 P St., Bldg 640 WPAFB, OH 45433-7765	Performing Organization Report Number AFIT/GCS/ENG/02-10	
Sponsoring/Monitoring Agency Name(s) and Address(es) sponsoring agency and address	Sponsor/Monitor's Acronym(s)	
	Sponsor/Monitor's Report Number(s)	
Distribution/Availability Statement Approved for public release, distribution unlimited		
Supplementary Notes		
Abstract		
Subject Terms		
Report Classification unclassified	Classification of this page unclassified	
Classification of Abstract unclassified	Limitation of Abstract UU	
Number of Pages 150		

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the United States Government.

AFIT/GCS/ENG/02-10

PERFORMANCE ANALYSIS OF A SECURE IEEE 802.11B WIRELESS
NETWORK INCORPORATING PERSONAL DIGITAL ASSISTANTS

THESIS

Presented to the Faculty of the Graduate School of Engineering and Management
of the Air Force Institute of Technology

Air University

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

John Lee Camp, B.S.

Captain, USAF

June, 2002

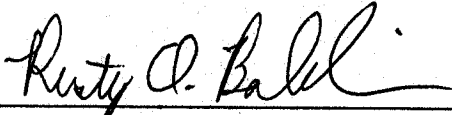
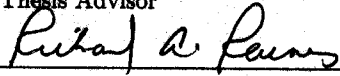
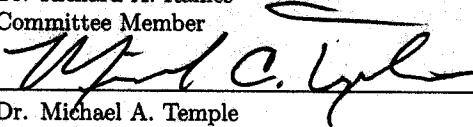
Approved for public release; distribution unlimited

PERFORMANCE ANALYSIS OF A SECURE IEEE 802.11B WIRELESS
NETWORK INCORPORATING PERSONAL DIGITAL ASSISTANTS

John Lee Camp, B.S.

Captain, USAF

Approved:

 _____	<u>18 Jun 02</u>
Maj Rusty O. Baldwin, PhD Thesis Advisor	Date
 _____	<u>18 Jun 02</u>
Dr. Richard A. Raines Committee Member	Date
 _____	<u>24 Jun 02</u>
Dr. Michael A. Temple Committee Member	Date

Acknowledgements

I would like to thank Major Rusty Baldwin for his support and guidance throughout my graduate work at AFIT. As an instructor and advisor he has worked to develop and expand my knowledge and experiences in Computer Systems. His patience and perseverance during the times when I had to shift focus back to my AFRL duties was instrumental in my success. I also extend my gratitude to Dr. Richard Raines and Dr. Michael Temple, their input and contributions were essential to my work.

I have been very lucky to have supervisors who have encouraged me and supported my academic endeavors. Thank you Dr. Michael Young, Mark Hoffman, and Michael Green for your cooperation and understanding.

Our computer support personnel, Dave Groomes, Chris Brooks, and Jeff Norton, have provided insight on the physical systems used. Their experience and knowledge in the configuration, setup, and testing of the research network used was invaluable.

I also want to thank my parents. They have developed and nurtured my curiosities and thirst for knowledge since I was a small child. The work ethic and dedication to excellence they have demonstrated is something I strive to apply to all I do.

I owe my deepest debt of gratitude to my wife and child. Without their love and inspiration, all my work is for nothing.

John Lee Camp

Table of Contents

	Page
Acknowledgements	iii
List of Figures	viii
List of Tables	ix
Abstract	xi
I. Introduction	1-1
II. Background	2-1
2.1 Introduction	2-1
2.2 The Personal Digital Assistant (PDA)	2-1
2.2.1 The Compaq iPAQ PDA Battery	2-2
2.3 Wireless Local Area Network (WLAN) Standards	2-3
2.3.1 Direct Sequence Spread Spectrum (DSSS)	2-4
2.3.2 Frequency Hopping Spread Spectrum (FHSS)	2-5
2.3.3 IEEE 802.11b Protocol	2-5
2.3.4 Bianchi 802.11 Model	2-7
2.3.5 Chhaya 802.11 Model	2-10
2.4 Network Security	2-11
2.4.1 Overview	2-11
2.4.2 Encryption	2-11
2.4.3 Firewalls	2-13
2.4.4 Authentication	2-13
2.4.5 Virtual Private Networks (VPNs)	2-14
2.4.6 Wired Equivalent Privacy (WEP)	2-17
2.4.7 WLAN Security	2-18
2.5 Summary	2-20

	Page
III. Experimental Methodology	3-1
3.1 Problem Definition	3-1
3.1.1 Goals and Hypothesis	3-1
3.1.2 Approach	3-1
3.2 System Boundaries	3-2
3.3 System Services	3-3
3.4 Performance Metrics	3-4
3.5 Parameters	3-4
3.5.1 System	3-4
3.5.2 Workload	3-4
3.6 Factors	3-5
3.7 Evaluation Technique	3-5
3.8 Workload	3-5
3.9 Experimental Design	3-6
3.9.1 Throughput Experiment	3-6
3.9.2 Battery Life Experiment	3-7
3.10 Summary	3-7
IV. Results	4-1
4.1 Throughput Experiment	4-1
4.1.1 VPN off	4-1
4.1.2 VPN On	4-7
4.1.3 Comparison	4-11
4.2 Battery Life Experiment Results	4-15
4.3 Summary	4-16

	Page
V. Conclusions	5-1
5.1 Results	5-1
5.1.1 Throughput Experiment	5-1
5.1.2 Battery Life Experiment	5-1
5.2 Conclusion	5-2
5.3 Summary	5-3
Appendix A. Transfer Program Code	A-1
A.1 Introduction	A-1
A.2 Server Side Code	A-1
A.2.1 <u>server.vbp</u>	A-1
A.2.2 <u>server.frm</u>	A-3
A.3 Client Side Code	A-38
A.3.1 <u>client.ebp</u>	A-38
A.3.2 <u>client.bas</u>	A-40
A.3.3 <u>client.ebf</u>	A-43
Appendix B. VPN-1 Configuration	B-1
Appendix C. Raw Data	C-1
C.1 Introduction	C-1
C.2 Throughput	C-1
C.2.1 Test 1	C-1
C.2.2 Test 2	C-2
C.2.3 Test 3	C-2
C.2.4 Test 4	C-3
C.2.5 Test 5	C-4
C.2.6 Test 6	C-4
C.2.7 Test 7	C-5

	Page
C.2.8 Test 8	C-6
C.2.9 Test 9	C-6
C.2.10 Test 10	C-7
C.2.11 Test 11	C-8
C.2.12 Test 12	C-9
C.2.13 Test 13	C-9
C.2.14 Test 14	C-10
C.2.15 Test 15	C-11
C.2.16 Test 16	C-11
C.2.17 Test 17	C-12
C.2.18 Test 18	C-13
C.3 Battery Life	C-14
Appendix D. Laptop Performance	D-1
Bibliography	BIB-1
Vita	VITA-1

List of Figures

Figure		Page
2.1.	Hidden Node Problem	2-6
2.2.	Bianchi Model Throughput Graph	2-9
2.3.	Chhaya Model Throughput Graph	2-11
2.4.	Data Encryption	2-12
2.5.	Network Topology	2-14
2.6.	Network Topology with Firewall	2-14
2.7.	VPN Network Topology	2-16
2.8.	VPN Packet	2-17
3.1.	System Under Test	3-3
4.1.	VPN On Throughput Decrease	4-12
4.2.	Client Distance Impact Graph	4-13
4.3.	File Size Impact Graph	4-14

List of Tables

Table		Page
2.1.	Compaq Constant Use Battery Life Testing (64 MB Color) (minutes)	2-3
3.1.	Experimental Factors	3-5
3.2.	Throughput Test Cases	3-6
3.3.	Battery Life Test Settings	3-7
4.1.	VPN Off Results	4-2
4.2.	VPN Off Computation of Effects	4-3
4.3.	VPN Off Interactions	4-3
4.4.	VPN Off ANOVA	4-4
4.5.	VPN Off Confidence Intervals for Effects	4-5
4.6.	VPN Off Confidence Intervals for Interactions	4-6
4.7.	VPN Off Client Distance Impact	4-7
4.8.	VPN Off File Size Impact	4-7
4.9.	VPN Off Factor Level Impact	4-7
4.10.	VPN On Results	4-8
4.11.	VPN On Computation of Effects	4-8
4.12.	VPN On Interactions	4-9
4.13.	VPN On ANOVA	4-9
4.14.	VPN On Confidence Intervals for Effects	4-9
4.15.	VPN On Confidence Intervals for Interactions	4-10
4.16.	VPN On Client Distance Impact	4-10
4.17.	VPN On File Size Impact	4-11
4.18.	VPN On Factor Level Impact	4-11
4.19.	Mean Throughput Comparison	4-12
4.20.	Client Distance Impact Comparison	4-13

Table		Page
4.21.	File Size Impact Comparison	4-13
4.22.	Battery Life duration Results	4-15
5.1.	Throughput Comparison	5-1
5.2.	Factor Impact	5-2
C.1.	Battery Life est Data	C-14
D.1.	Laptop Results	D-1
D.2.	Laptop Computation of Effects	D-2
D.3.	Laptop Interactions	D-2
D.4.	Laptop ANOVA	D-2
D.5.	Laptop Confidence Intervals for Effects	D-2
D.6.	Laptop Client Distance Impact	D-3
D.7.	Laptop Workload Impact	D-3

Abstract

Research results of this indicate very poor performance of a Wireless Local Area Network (WLAN) utilizing PDAs. Network throughput is adversely effected most by VPN implementation and slightly by increased file size. The client distance factor has virtually no effect on the throughput. The impact of each of these factor levels is small when compared to the magnitude of the overall mean throughput ($\leq 6\%$). The average network throughput with the PDA client is much lower than expected ($\approx 11,500$ bps). This is attributed to several factors with degradation primarily resulting from limitations of the PDA hardware and O/S. Because of the low throughput values achieved (regardless if VPN is off or on), an operational WLAN with PDAs (as tested) is not feasible. Operational use of the network tested would require an in-depth analysis of the type of network traffic and performance required to maintain functionality. To deploy such a system, custom designed Winsock controls would need to be implemented to minimize limitations imposed by the PDA. As PDA technology continues to develop, future hardware and O/S functionality may provide a more robust platform for network communications. The battery life of the PDA and jacket battery combination is observed to be about 164 minutes with additional jackets adding about 99 minutes each.

PERFORMANCE ANALYSIS OF A SECURE IEEE 802.11B WIRELESS NETWORK INCORPORATING PERSONAL DIGITAL ASSISTANTS

I. Introduction

As technology advances and the power of hand-held computing devices increases, the demand for mobile Wireless Local-Area-Networks (WLANs) has grown. These networks typically consist of several host desktop computers and can include any number of mobile devices, including laptop computers and PDAs (Personal Digital Assistant). Desktop computers sometimes serve as stable, stationary reach back machines providing a connection to larger networks. To maintain connectivity throughout the range of these devices, a wireless network is necessary. While current laptops provide performance comparable to that of current desktops, hand-held devices such as PDAs are not nearly as sophisticated, i.e., they lack the raw processing power and large memories of desktop computers. Such constraints limit the size and complexity of their software.

Hardware and software are not the only difficult issues involved with wireless network devices. Due to the unguided medium of wireless networks, they are not as reliable as wired networks transmitting over guided media such as optical fiber or copper wire. More overhead bits are transmitted, routing information is more complex and correction codes are more intricate due to higher bit error rates caused by increased noise and signal degradation. Transmitted signals may be received multiple times because of multipath effects. Therefore, receivers must be robust enough to effectively distinguish and correct for this. All these effects decrease the efficiency of wireless data transmission.

Another difficulty encountered in wireless networks is security. The inefficiency of data transmission, limited computing power, and limited battery life of hand-held devices make it challenging to efficiently implement secure communications. Sending packets through the atmosphere makes

them available to anyone within range of the transmitter. Since wireless transmissions are so easily intercepted, a robust security system is absolutely necessary. Traditional complex encryption software cannot be hosted on PDAs due to limited resources, and the inefficiencies of data transmission limits the use of long, complex encryptions which increase packet size. While several industry standards exist for wireless LAN protocols, none of them adequately address security, specifically, encryption protocols. A Virtual Private Network (VPN) system provides many of these requirements in a single package. VPNs provide privacy and security through software hosted on the client as well as the reachback computer. Recent advances in PDA programming have led to compact PDA versions of VPN software [Usk97, Kor98, Ven01, WIASG01, You00, WIASG01].

The Sustainment Logistics Branch (Deployment Sustainment Division, Human Effectiveness Directorate) of the Air Force Research Laboratory has developed several intelligent agent programs. One is the Human Interaction with Software Agents (HISA)[MW02]. HISA provides air-traffic controllers with detailed information about all current planes of interest to the controller. The software provides information such as flight paths, weather updates, and maximum aircraft on ground data for target airfields. As a flight progresses, information about the aircraft is regularly updated. When certain warning criteria are met, the software sends an alert to the controller. For example, if an aircraft's projected flight path intersects with another aircraft or weather system, the controller will be notified and a candidate solution provided. Controllers then decide to accept the candidate solution or create their own. This software provides a powerful way to track information in a user-friendly manner for the controller.

Such software is currently only available on desktop computers but there is great operational potential in putting the software on PDAs. By placing the software on hand-held devices, flight-line (or otherwise mobile) controllers could monitor vital information and receive updates without being tied to a desk. Several mobile devices could communicate with a host desktop, regularly sending and receiving updates. Updates will be bursty in nature. Obviously, the information

passed during updates is sensitive so broadcasts must be secure. To safely accomplish this, a secure wireless LAN is necessary. The primary goal of this research is to determine if it is feasible from a performance perspective for such applications to use secure wireless LAN's implementing current wireless protocols and VPN software?

Chapter II reviews current literature on the topic. A background discussion on PDAs, security, and WLANs is given, as well as a discussion of relevant articles and papers. An overview of the current state of research in this area is also included, demonstrating the applicability and relevance of this work.

Research methodology is presented in Chapter III. Specific goals are described and a hypothesis given. The approach for meeting these goals and determining the validity of the hypothesis is presented as well.

Chapter IV presents the research results. Detailed accounts of the throughput and battery life test data are given. The information is summarized and an analysis given to provide a clear indication of the performance of the WLAN, PDA, and VPN software. The throughput experiment and battery life experiment are discussed separately.

The significance of the results are explained in Chapter V. Results are summarized and a conclusion is given. Some implications and the research impact is provided along with recommendations for future uses of the system tested along with possible topics for further research in this area.

II. Background

This chapter reviews current literature on the research topic. Background on Personal Digital Assistants (PDAs), security, and WLANs is given, as well as a discussion of relevant articles and papers. An overview of the current state of research in this area is also included, demonstrating the applicability and relevance of this work.

2.1 Introduction

Wireless local area networks (WLANs) are an exciting growing technology. The use of Personal Digital Assistants (PDAs) is also on the rise. The merging of these two technologies combined with the need for security has created research opportunities in many fields. The purpose of this chapter is to describe some of the research involving WLANs and wireless security.

2.2 The Personal Digital Assistant (PDA)

The presence of PDAs is growing very rapidly. They are being used by housewives and C.E.O.s [Fre02]. They can hold calendar, address book, and task information as well as corporate data. Applications are being developed to provide a wide range of capabilities on these little computers. These devices run on operating systems that are scaled down versions of full-size computer operating systems (O/S). The O/S is minimized to create the smallest possible memory requirements and still provide the necessary capabilities. A stable storage ROM device provides basic programs for the PDA. User data and applications are stored in small, volatile RAM, in some cases Flash memory is used which provides data stability. The displays are LCD screens that are available in 16 bit grayscale to 64 MB color. Input is touchscreen using a stylus or through external, collapsible, compact keyboards. Data synchronization, generally through Universal Serial Bus (USB) ports, is available with most desktop computers. This provides a way to move files and data to and retrieve it from the PDA, and can also be used for application installation onto the PDA. Most PDAs do not

have the inherit capability to interact with add-on cards, an extension jacket is necessary. Compaq has two types of jackets, one provides access to accessory cards that have a Compaq proprietary interface, the other contains an additional jacket and provides access to standard PC cards [Fre02].

Any security program or tool designed for WLANs and more specifically, PDAs, must be flexible and very efficient. PDAs generally contain only 1 to 10 percent of the memory commonly available on desktops computers and are generations behind in processor speed [Fre02]. While most state-of-the-art PDAs now offer add-on memory cards for data memory, the majority of data resides in the PDAs main memory. Power consumption is a major issue since the PDAs must run off small, limited capacity batteries. The throughput of PDA network cards is another major limiting factor. As opposed to wired networks that have low bit error rates, WLANs have bit error rates orders of magnitude greater. In general, as the reliability and security of the network improves, data throughput decreases [YF00, Rus01, Fre02].

2.2.1 The Compaq iPAQ PDA Battery. Battery design and functional information as well as usage scenarios and power management suggestions from Compaq on iPAQ series H3100, H3600, H3700 PDAs and the Compaq PC card expansion jacket is provided in [Vin01]. The internal PDA battery has a 1000 mAh capacity, which is the same as the expansion jacket. The PDA battery function has been designed so that usage can only drain the PDAs battery to a certain level. A minimum amount of the charge is maintained in order to preserve user data and applications. The jacket battery cannot directly power the PDA, but once the PDA battery decreases to a level below that of the jacket battery, the jacket will begin to charge the PDA. The PDA cannot charge the jacket. Since the jacket battery charge can be fully used, the additional jacket battery has a greater usable capacity than the PDAs internal battery, even though they have the same battery capacity rating. PC card accessories have a dramatic effect on the battery life duration, especially with cards that are not designed specifically for PDAs. The Constant Use Battery Life Testing table

in [Vin01] provides results on Compaq's battery testing and is replicated below in Table 2.1. The unit used in this research is 64 MB Color Model.

Table 2.1 Compaq Constant Use Battery Life Testing (64 MB Color) (minutes)

Brightness/ Audio Volume Setting	iPAQ	iPAQ w/ PC Card Expansion Jacket Empty	iPAQ w/ PC Card Expansion Jacket (avg of cards tested)
Super/Loud (6/6)	90 - 105	330 - 345	180 - 195
Super/Disabled	105 - 120	405 - 420	195 - 210
High/Middle (3/6)	105 - 120	360 - 375	195 - 210
Medium/Disabled	135 - 150	465 - 480	210 - 225
Low/Silent (1/6)	135 - 165	435 - 450	180 - 195
Low/Disabled	165 - 180	525 - 540	210 - 225
Power Save/Loud (6/6)	345 - 360	915 - 930	270 - 285
Power Save/Silent (1/6)	360 - 375	975 - 990	375 - 390
Power Save/Disabled	540 - 555	1115 - 1130	330 - 345

2.3 Wireless Local Area Network (WLAN) Standards

WLANs are of two major types, Infrared (IR) and Radio Frequency (RF). IR networks are used for short-range open areas and are primarily line of sight. RF networks offer much greater ranges and can be used in larger networks. This research will focus on the RF networks [MM99, Kor98].

WLANs are popular due to the ease of installation and inherit mobility of the systems. WLANs do not require the wired backbone support of a wired network. They are quick to install and never require upgrades to current wiring or dealing with the installation of a new wired network. PDA clients can range up to a quarter of mile away from a receiver and still transmit effectively. Strategically placed network bridges and antennas can extend the range between the client and wired network indefinitely. Unfortunately, this mobility and flexibility leads to many of the problems for the networks. While wired networks conduct communications in specific and orderly ways, WLANs have very open and irregular communications. In a wired network, data packets travel through a guided media and can only be accessed by tapping into this media. Wireless transmissions occur in the open atmosphere, where they can be naturally reflected and duplicated, resulting

in numerous “copies” of each transmission, all readily available to anyone interested. WLANs enhance many existing applications and make others possible, such as warehouse inventory taking, co-operative learning, and mobile system monitoring. They also generate advanced requirements to meet technical and regulatory specifications. Many systems operate in a mixed network environment that combines both WLANs and traditional LANs. The network backbone may consist of a wired network, while the end connections may be wireless [MM99, KK00].

Most WLANs operate in the Industrial, Scientific, and Medical (ISM) bands in the United States. The frequencies of these bands are: 902-9289 MHz, 2.4-2.4853 GHz, and 5.728-5.85 GHz. Operating in these bands does not require licensing by the Federal Communications Commission (FCC), but does require the use of spread spectrum technology and limits transmit power to one watt EIRP (Effective Isotropic Radiated Power). The two most commonly used spread spectrum technologies are Direct Sequence Spread Spectrum (DSSS) and Frequency Hopping Spread Spectrum (FHSS) [MM99].

2.3.1 Direct Sequence Spread Spectrum (DSSS). Many state-of-the-art WLANs use direct sequence spread spectrum. In DSSS, the original signal is spread across a given bandwidth using a spreading code. This code, called the spreading sequence, consists of binary digits that modulate the original signal. The ratio of the number of bits in the spreading code per data bit sent is called the spreading ratio. Determining an efficient and effective spreading ratio is a balancing act; a lower ratio makes more efficient use of the transmitting band, while a higher ratio makes the signal more resistant to noise. As long as the signal is sufficiently spread, interference in a portion of the frequency band can be tolerated without inducing bit errors. Without knowledge of the spreading ratio and sequence it can be difficult to intercept and interpret DSSS signals; but if the equipment being used is known, transmitted information can be obtained using similar equipment and scanning across the channels [MM99, SBB01, Rus01].

2.3.2 Frequency Hopping Spread Spectrum (FHSS). While frequency hopping spread spectrum is not quite as common as direct sequence spread spectrum, it is used in many systems. As the name suggests, FHSS takes the original signal and transmits some data at a one frequency, then “hops” to another frequency to transmit. The amount of time spent at a frequency is called the dwell time and the individual frequencies are called subchannels. In order to function properly, the transmitter and receiver hop sequence must be synchronized before transmission begins [MM99, Rus01].

2.3.3 IEEE 802.11b Protocol. The IEEE 802.11b standard defines a Medium Access Control (MAC) layer and several Physical Layers (PHY): DSSS, FHSS, and IR. It also defines a privacy standard called Wired Equivalent Privacy (WEP). An 802.11b network may consist of several types of stations. These stations types are fixed, portable (movable, but used at a standard location), or mobile. Two or more stations communicating together create a Basic Service Set (BSS). An isolated BSS that does not connect to a larger wired network (e.g., via a base computer) is considered an Independent Basic Service Set (IBSS). A Distribution System (DS) is used to connect multiple BSS’s. The BSS’s access the DS through an access point, which is an addressable station. Larger networks consisting of many DS’s and BSS’s are called Extended Service Sets (ESS). An ESS is robust enough to support movement of stations throughout the ESS, between separate BSS’s [MM99, Bia00].

Most wired LAN’s use Carrier Sense Multiple Access with Collision Detection (CSMA/CD) to resolve packet collisions. WLANs use CSMA with collision avoidance or CSMA/CA since radios cannot transmit and receive simultaneously with a single antenna. This limitation results in the “hidden node” problem depicted in Figure 2.1. Because nodes A and C cannot detect each other’s transmissions, they may attempt to simultaneously send a message to node B located between them resulting in a collision and data loss. Node B can “hear” both nodes A and C, yet nodes A and C cannot sense each others transmission.

Methods used by 802.11 to overcome the hidden node problem are discussed later. Collision avoidance in 802.11b is implemented using what is called the Distributed Coordination Function (DCF) [MM99, Bia00]. The DCF uses an exponential backoff counter to deconflict channel access after a collision. When a collision or failed transmission occurs, the sending node will wait a number of slots, from 0 to $W-1$ before resending. The variable W is called the contention window. The minimum contention window is denoted as W_{min} and the maximum contention window is denoted as W_{max} . The two are related by the following equation: $W_{max} = 2^m W_{min}$, where m is maximum number of times the node can try to resend. The contention window, W , begins at a default minimum, then doubles after each collision or failure until it reaches the default maximum. The default minimum and maximum for FHSS are 16 and 1024, for DSSS are 32, and 1024 slots respectively [MM99, Bia00].

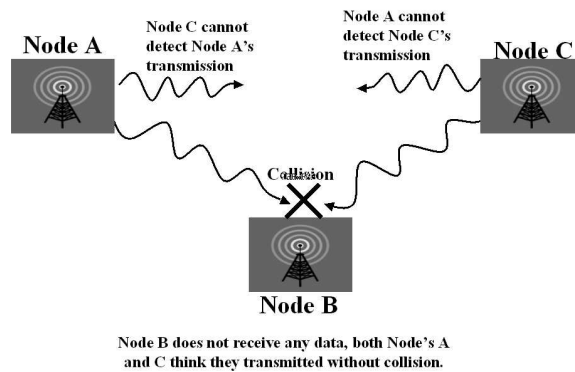


Figure 2.1 Hidden Node Problem

The DCF has two methods to coordinate packet transmission. The default service is the basic access mechanism. In this scheme, a node that is waiting to transmit will decrement its backoff counter while channel is idle. If a transmission is detected, the waiting node suspends decrementing the counter until the channel is idle for a period of time equal to a Distributed InterFrame Space (DIFS), also measured in slots. When the DIFS period has elapsed without the channel being busy, the counter will continue to be decremented. If the channel is idle when the counter reaches

zero, the node will transmit. Once a destination node has successfully received the packet, it will immediately send an acknowledgment message to the sending node. If, however, the sending node does not receive the acknowledgment within a set amount of time, it will resend the message [MM99, Bia00].

Since the default scheme does not address the hidden node problem, an optional four-way handshake technique is also supported. This is the Request-To-Send/Clear-To-Send (RTS/CTS) mechanism. This mechanism listens to the channel and waits when it is busy just like the default mechanism; but when the backoff counter reaches zero, a Ready-To-Send message (RTS) is sent instead of a data packet. The RTS is a short message that contains the destination address as well as the amount of time needed to send the message. If the receiving node receives the RTS without error, it replies with a Clear-To-Send (CTS). Once the CTS is received, the sender transmits the packets. Other nodes that receive the RTS and CTS know that the channel is busy. The receiver sends an acknowledgment for each packet. If the sender does not receive the acknowledgment, it resends the packet. This helps solve the hidden node problem because the sending nodes only send small RTS messages before receiving a CTS message. Because of the small size of the RTS messages, RTS collisions are rare. If a collision between two RTS messages does occur, the cost is not very great; the time spent creating and sending the message is very short compared to the time spent sending data packets. The CTS message serves to prevent data packet collisions, all nodes within sending distance of the receiving node receive the CTS message and know not to transmit. In the example of Figure 2.1, suppose node A sends a RTS message. When node B receives this message it replies with a CTS message. Both nodes A and C receive the CTS and node C knows not to transmit. Even though node C does not directly know that node A is transmitting, it knows that it cannot transmit because of the CTS message [MM99, Bia00].

2.3.4 Bianchi 802.11 Model. Giuseppe Bianchi created a detailed and well documented analytical model of the 802.11 (RTS/CTS) protocol described in [Bia00]. Bianchi examines the

expected performance of a network, looking at the time spent sending handshake and coordination messages, as well as time lost due to idle and collision times. By calculating the time spent actually sending data bits and comparing that to the total time associated with that transfer, the efficiency of the network is modeled. In this model the throughput, S , is defined as [Bia00]

$$S = \frac{E[P]}{T_s - T_c + \frac{\sigma(1-P_{tr})/P_{tr} + T_c}{P_s}} \quad (2.1)$$

where P_s is the probability the transmission on the channel is a success, P_{tr} is the probability that there is at least one transmission in the time slot, $E[P]$ is the expected packet payload size, σ is the duration of an empty time slot, T_s is the average time the channel is sensed busy because of a successful transmission, and T_c is the average time the channel is sensed busy because of a collision. P_s , P_{tr} , T_s , and T_c , can be found using the following formulas [Bia00]

$$P_s = \frac{n\tau(1-\tau)^{n-1}}{1-(1-\tau)^n}, \quad (2.2)$$

$$P_{tr} = 1 - (1-\tau)^n, \quad (2.3)$$

$$T_c = RTS + 3SIFS + 4\delta + CTS + H + E[P] + ACK + DIFS, \text{ and} \quad (2.4)$$

$$T_s = RTS + DISF + \delta \quad (2.5)$$

where n is the number stations, τ is the probability that a station transmits a packet, RTS is the transfer time for the RTS message, $SIFS$ is the time of the SIFS period, δ is the propagation delay, CTS is the transfer time for the CTS message, H is the time overhead created by the physical layer header plus the MAC layer header, ACK is the transfer time for the ACK message, and $DIFS$ is the time of the DIFS period. The probability that a station transmits, τ , is given by the formula [Bia00]

$$\tau = \frac{2(1-2p)(1-p)}{(1-2p)(W+1) + pW(1-(2p)^m)} \quad (2.6)$$

with

$$p = 1 - (1 - \tau)^{n-1} \quad (2.7)$$

where p is the probability of collision, W is the value of the backoff counter, and m is the maximum backoff value.

Figure 2.2 shows the predicted throughput using this model using the following parameter values: $E[P] = 8184$ bits, $H = 400$ bits, ACK = 240 bits, RTS = 188 bits, CTS = 240 bits, $\delta = 1 \mu s$, $\sigma = 50 \mu s$, SIFS = $28 \mu s$, and DIFS = $128 \mu s$. The graph consists of four curves, each for different numbers of stations. The throughput, S , is plotted against the transmission probability, τ . The maximum value for these curves is approximately 0.84 and occurs at $\tau = 0.02$ for $n = 5$, 0.01 for 10, 0.005 for 20, and 0.002 for 50. As the number of stations increase, S reaches its maximum quicker, but drops off at a much faster rate than the smaller networks. The maximum occurs sooner because the network has higher utilization from more stations, and the rapid drop occurs because of the increase number of collisions [Bia00].

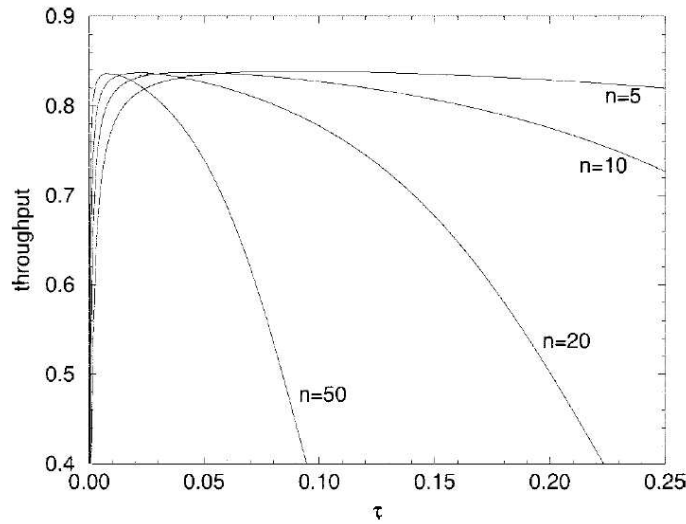


Figure 2.2 Bianchi Model Throughput Graph

2.3.5 *Chhaya 802.11 Model.* Harshal Chhaya and Sanjay Gupta have also developed an analytical model of the 802.11 (RTS/CTS) protocol in [CG96]. This model provides a formula to determine network throughput. This throughput value, S , represents the number of successful packet transmissions over a given interval of time. A statistical definition of this time, the renewal interval, is used to calculate S . Throughput, S , is defined as [CG96]:

$$S = \sum_{i,j \in A} S(i,j) \quad (2.8)$$

and

$$S(i,j) = \frac{\frac{p_s(i,j)}{G} [G(i,j) + (1 - e^{-\beta G(i,j)}) \sum_{(m,n) \in C(i,j)} G(m,n)]}{1/G + l + l_{rts} + l_{cts} + 3 * SIFS + DIFS + \beta + l_{ack}} \quad (2.9)$$

where $S(i,j)$ is the throughput from node i to node j , A is the set of all nodes in the network area, $p_s(i,j)$ is the probability that the transmission from node i to j is successful, $G(i,j)$ is the rate of the exponential distribution that defines the time until a transmission is generated at i destined for j , G is the sum of the rates for all the node pairs in A , l is the length (in time units) of a data frame, l_{RTS} , l_{CTS} , l_{ACK} : The length (in time units) of a rts, cts, and ack frame normalized by the expected length of a data frame, β , is the propagation delay normalized by the expected length of a data frame, $DIFS$ is the time of the DIFS period, and $SIFS$ is the time of the SIFS period.

Using the above formula with the following typical variables: $l = 1, \beta = 0.05, SIFS = 0.05, DIFS = 0.15, l_{ack} = 0.15$, and $l_{rts} = l_{cts} = 0.15$, and a randomly selected set of nodes with various coordinates, results in Figure 2.3 [CG96]. The number, nature, and location of nodes was used to determine the remaining factors.

This graph shows the throughput value, S , as a function of the offered load, G . Several curves are shown representing S with a variety in the number of nodes in the network. The results of this model are similar to those of the Bianchi model, but not quite as limited. For these plots, the maximum occurs with G between 5 and 10. After the maximum is reached, S decreases, but even

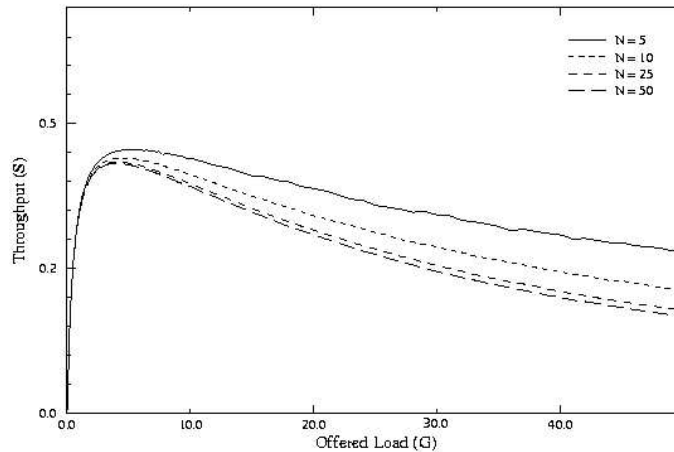


Figure 2.3 Chhaya Model Throughput Graph

for $N = 50$, this drop is gradual. Again, an initial increase in load raises S to the maximum, but as the load increases, S decreases due to more collisions [CG96].

2.4 Network Security

2.4.1 Overview. Computer network security incorporates many possible capability options. There are various levels of security that can be applied and different methods to achieve the desired ends. Security can consist of data encryption, network isolation, user authentication, or any combination of these. This section provides basic information about these security capabilities and their usefulness and impact on a WLAN.

2.4.2 Encryption. A basic form of security, data encryption consists of taking user data and manipulating it in some known manner. Ideally, only the data source knows exactly how the data is encrypted, and only the source and destination know how to decrypt it. Take the data packet in Figure 2.4 as an example. The user application on computer one has created data that needs to be sent to computer two. Computers one and two established a trust relationship at some time in the past and have knowledge of expected key usage. The application data along with any of the upper layer headers and trailers is encrypted and unreadable without the decryption

key. Encryption is accomplished through a series of calculations based on a specific algorithm and encryption key. The encrypted data is packaged for transmission over the network with appropriate network and physical layer headers and trailers, then sent to computer two. Computer two knows which decryption key to use to decrypt data from computer one. In this manner, data can be sent between two computers without compromising the data [Lab00, Sti95].

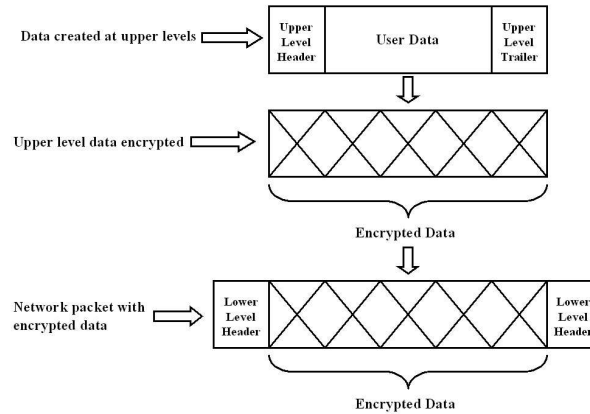


Figure 2.4 Data Encryption

To ensure security, modern encryption algorithms are highly scrutinized. These algorithms were designed primarily for use on wired networks where data routing and accessibility is more controlled than a wireless environment. Encryption is only a data protection scheme, network control and protection (i.e., authenticated) must be provided through other means [Sti95].

Three widely used algorithms in wired networks are RSA and Triple-DES (Data Encryption Standard) [Lab00] for encryption and Kerberos [KK00, NT94, Sti95] for authentication. All are computationally expensive and are not well suited for WLANs. The symmetric key cryptography of triple-DES is quite sophisticated and maintains security without exposing individual keys and can be quite fast when implemented in hardware. RSA, however, creates a great deal of overhead and is relatively slow. When implemented in a wireless network, efficiency and throughput tend to fall to unacceptable levels [KK00]. Symmetric key algorithms are less computationally complex and quicker but require complicated key control and distribution methods that work against the

ease and flexibility of a WLAN. Kerberos also requires undesirable levels of overhead. To ensure authentication, six handshake messages must occur between the client and three separate server machines [NT94]. WLAN security and encryption has the dual challenge of requiring more flexibility and yet also be more robust than wired network counterpart [KK00, NT94].

2.4.3 Firewalls. Network isolation is a very popular method for achieving security. By creating a boundary, or firewall, at the edge of a network, all incoming and outgoing traffic can be controlled. Figure 2.5 shows a typical network topology that is not isolated. Switches connect workstations and servers through a router. The router also provides access to an external network, in this case, the Internet. All computers in the network have free access to the Internet and vice versa. Figure 2.6 shows a network with a firewall in place. With this implementation, all network traffic behind the firewall is unrestricted. Local machines still have complete access to one another. However, any traffic that must leave or enter the local network goes through the firewall server, the single point of access to the external network. This single controlled entry point creates a potential bottleneck in network traffic. The source and destination information is masked at the firewall so external computers do not receive information about the internal network. Incoming messages can be screened and monitored to implement control mechanisms when necessary.

One example of a firewall system consists of a server with routers on either side. Because all external traffic will pass through this computer, it must be powerful and reliable enough to manage large amounts of data. Routers typically provide initial screening of all packets based on a set of static rules. The server offers more flexible control and oversight of the routers. All transmissions and messages can be logged. By restricting access through this single point, the firewall system can carefully guard and inspect all traffic flowing to and from the LAN [MBWV98, Usk97].

2.4.4 Authentication. Often used in conjunction with a firewall, authentication is an access control mechanism for users and computers on the network. An authentication server maintains information on all authorized users and devices. When a user logs in, they are required to pro-

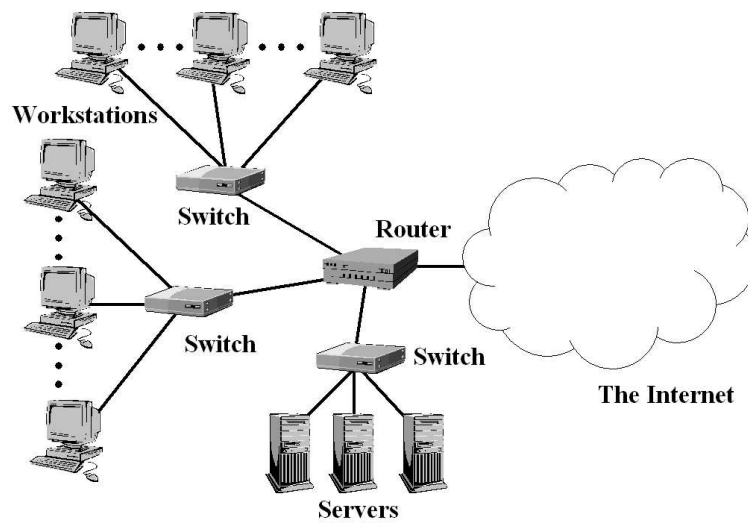


Figure 2.5 Network Topology

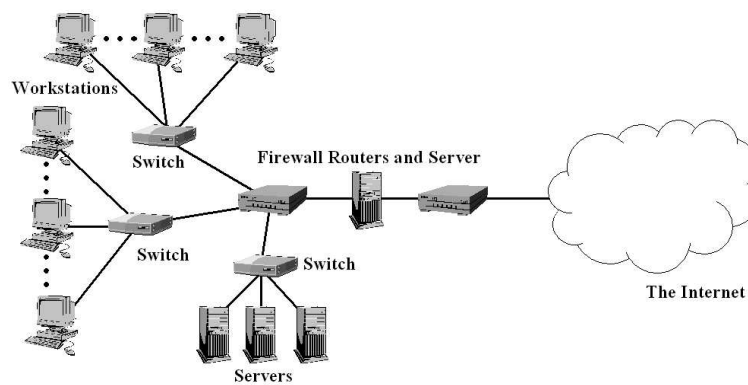


Figure 2.6 Network Topology with Firewall

vide a password and account ID. Devices are often monitored through their unique MAC address. Authentication provides confidence that everything participating in the network is a validated and allowed entity.

2.4.5 Virtual Private Networks (VPNs). While encryption, firewalls, and authentication provide key aspects of network security, a Virtual Private Network, or VPN, provides all this in a single package. VPNs provide improvements to security that are needed before WLANs are allowed connectivity to sensitive military or other government wired networks. A VPN is a combination

of tunneling, encryption, authentication, and access control used to carry traffic over an unsecure network. It is implemented by using a portion of a shared wired network's bandwidth to emulate the characteristics of a private network. These characteristics are: to provide the ability to exchange information between separated pieces of the network and to provide the privacy and security of a true private network. The connection is virtual; the VPN topology is constructed on top of shared physical network. A VPN provides robust encryption and authentication protocols enabling secure transmissions between two separate, secure networks over an unsecure network. Originally, VPNs were designed for communications over a wired network; but the same security requirements exist in WLANs. Instead of creating a virtual connection over shared wire, a VPN can be used to create a virtual connection across shared airspace. Packages are bundled and encapsulated, then passed through a set of nodes specified by the routing headers attached to the newly formed package. Encryption is used not only on data, but on routing information as well. Only the minimum amount of information necessary to get the encrypted packet from one firewall to the other is left in the clear. Firewall servers in VPNs, often called gateways, are responsible for examining all traffic into and out of the network. [Usk97, Kor98, Ven01, PE00, You00, WIASG01].

There are two primary tunneling protocols in use for VPNs, both developed by the Internet Engineering Task Force (IETF - www.ietf.org). The first is a layer 2 tunneling protocol (L2TP) [Ven01, You00]. This data link layer protocol is beneficial because it supports the Point-to-Point Protocol (PPP) for encapsulation which does not require additional software for the client. The second protocol is the Internet Protocol Security (IPSec), a level 3 network layer implementation. This protocol provides end-to-end IP traffic security and was designed specifically for larger, wired networks. Neither of these protocols are fully mature; they are being continuously improved and further developed [Usk97, Kor98, Ven01, Cus01, You00].

VPNs offer a range of encryption and authentication protocols adjustable to meet various security requirements. The encryption algorithms run from the weak privacy of WEP to the

relatively strong triple-DES encryption [Ven01, You00]. Authentication methods include MAC address approval tables, key-enabled authentication, as well as authentication headers that provide verification of the integrity of received packets. While VPNs provide required security features, current implementations can drastically degrade throughput. In [PE00], a high security VPN caused up to a 65 percent decrease in data throughput on a 100 Mbps network, and 97 percent utilization of an AMD K6 400 MHz PC. The degradation is not nearly as severe in lower speed networks (33.6 Kb/s) [Usk97, Kor98, Ven01, PE00, You00].

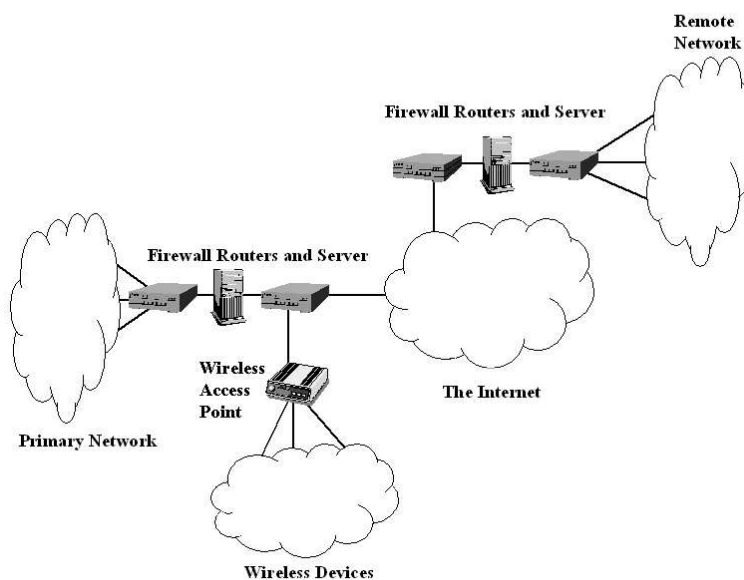


Figure 2.7 VPN Network Topology

VPNs incorporate aspects of all the previously mentioned security capabilities into one package that provides more comprehensive security. Figure 2.7 shows one possible VPN setup. This VPN consists of three separate networks, the primary, remote, and wireless networks. Data sent from any of these networks to another is encrypted. When two computers within either the remote or primary network need to communicate, it is done without any encryption. Since user and device authentication are in place, two computers behind the same firewall are trusted and can

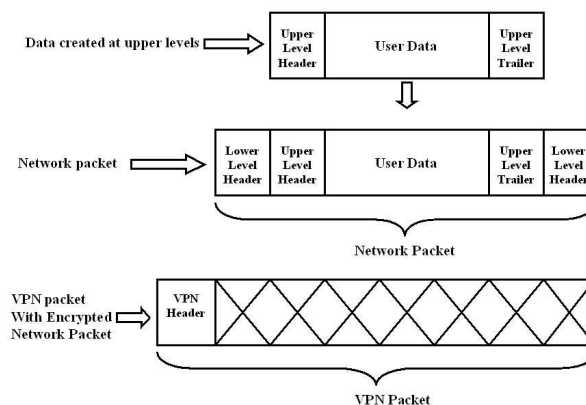


Figure 2.8 VPN Packet

communicate freely. When computers from the primary and remote network need to communicate, the procedures are very different. Assume computer one, in the primary network, needs to send data to computer 2, in the remote network. Computer one sends an unencrypted message to the firewall (or gateway) server which examines the source and destination of the packet, then encrypts it (Figure 2.8). This encrypted packet has a header that contains the minimum amount of information necessary to get the packet to the firewall of the remote network. When the VPN is initially established, communication between the firewall servers is conducted to develop the proper communication paths and methods. The receiving firewall examines the incoming packet and decrypts the packet. The decrypted packet is sent, within the remote network to its destination.

The Air Force Systems Networking Program Office at Maxwell AFB, AL has been working on an Air Force wide VPN implementation, the Common User (CU) Virtual Private Network. The goal of this program is to provide information protection of its sensitive but unclassified data transmitted over the DoD Intranet (NIPERNET) [Off01]. This VPN will provide secure communications between all AF installations. Testing of this VPN with a gateway server with one network card has demonstrated a the maximum throughput of 70 Mbps [Off01].

2.4.6 Wired Equivalent Privacy (WEP). The only inherent 802.11b data protection scheme is Wired Equivalent Privacy. This scheme was not intended to provide a strong encryption

of data, only to offer a measure of privacy that one would expect on a wired network. WEP currently has two different forms, a 40-bit key weak encryption; and a newer 128-bit key encryption [BB01, SBB01, Rus01, Usk97, Wea00, Kor98, SIR01].

While this scheme provides some privacy, it does not provide adequate security. WEP uses an RC4 [Lab00] cipher to encrypt data. Unfortunately the same secret key is used for all devices on the network and also has very predictable methods for generating an initialization vector, which is attached to the secret key and changed with every use [SIR01]. These two characteristics make it easy to break. A possible solution for this problem is manual generation and maintenance of all keys. While effective, this is support intensive and not practical for large networks. According to 1995 figures, the 40-bit encryption can be broken through brute force methods in 2 seconds with hardware worth \$100,000 and in merely 2 ms with hardware worth \$1,000,000 [Usk97]. Since 1995 the price of hardware has dropped drastically. It is not practical to break the 128-bit version through brute force, but the key-based implementation still leaves a WEP-only protected network vulnerable. A single static key is typically used for the entire network. If this single key is compromised, the entire network is compromised [BB01, SBB01, Rus01, Usk97, Wea00, Kor98, SIR01].

2.4.7 WLAN Security. The ability to safeguard a wireless network is difficult and 802.11b's inherit privacy scheme, WEP, is not robust enough. Other security measures must be used to protect wireless networks. A popular method used to improve security is to implement a secure mutual authentication process. In this process, a wireless node is denied access to the wired network until the wireless node and wired access point can mutually authenticate. During this authentication, a secure session key is created between the access point and wireless node. After authentication, the session key is used to encrypt and send the WEP encryption key. This type of mutual authentication is handled differently in various implementations, but the basic idea is the same. Another method is to use controlled lists of approved MAC addresses. Any MAC address

not approved will not be authenticated. This scheme can only be realistically implemented a small, controlled, relatively static network [BB01, SBB01, Rus01, Usk97, Kor98, Gar02].

Other security concerns revolve around the “free” nature of the wireless transmissions. Authentic signals are easily intercepted and available for examination. Ambient noise signals are abundant and need to be sorted through. Signals not intended for the network and other malicious signals must be rejected. One problem is interference created from other products using the same frequency band. Another problem is denial-service attacks that hackers can generate by saturating the network with jamming signals. Because it must be assumed that authentic signals will be intercepted, they need to be encrypted or otherwise safeguarded. WLANs must be intelligent enough to recognize signals from their own network and ignore all others. Furthermore, hackers often use specialized transmitters and receivers to deceive authentic stations. By mimicking an authentic base station, a hacker can route all legitimate traffic through their own system, quietly monitoring all traffic; possibly gaining authentication and communication protocol information. By mimicking an authentic mobile station, a hacker may be able to convince the network to allow the hacker access. Once the fake station gains access, the entire network is compromised [MBWV98, Rus01, Usk97, Gar02].

A source of concern with PDAs is their physical size and accessibility. The inherent mobility of a PDA leads to theft susceptibility. Careless users may set the PDA down for just a minute, or possibly forget the PDA somewhere. If they were logged in at the time, a hacker has free access to the PDA and the network. Even if the user is not logged in, native password control on PDAs is very weak and easily defeated. Due to the lack of permanent storage, PDAs store all data in their RAM, which makes them very vulnerable to virus and trojan horses [Gar02].

The current trend in WLAN security is a movement to VPN implementation [Ven01, You00, WIASG01, Off01, Gar02]. Recent releases of commercial VPNs include PDA client software that

claims to be efficient, yet powerful, making WLAN VPNs possible. These VPNs may provide the necessary security, but is the performance impact too great [Ven01, You00, WIASG01, Gar02]?

2.5 Summary

This chapter provides an overview of WLAN and PDA technology and protocols. In-depth information about the wide range of security concerns and issues is presented, along with possible solutions and ways to combat the problems. Research into WLANs is covered, providing insight to the analysis of the WLANs.

III. Experimental Methodology

This chapter contains the methodology used in this research. Specific goals are described and a hypothesis given. The approach for meeting these goals and determining the validity of the hypothesis is presented as well.

3.1 Problem Definition

3.1.1 Goals and Hypothesis. This research is divided into two major sections, a throughput performance analysis and a battery life duration determination. The primary goal of the throughput performance analysis is to evaluate the effect of a virtual private network (VPN) implementation on the throughput of a wireless local area network (WLAN) which contains personal digital assistant (PDA) clients. The secondary goals are to evaluate the effect on throughput from the size of transferred files and client distance from the receiving antenna. The goal of battery life duration experiments is to determine how long the PDA can operate under constant use as well as to estimate the amount of time the battery in the external jacket can extend the life of the unit.

It is expected that the throughput of the WLAN will decrease under the following conditions: implementation of the VPN, increased file size, and increased client distance. The VPN is expected to have the greatest effect on the throughput, much greater than both the file size and client distance. Based on information from the Compaq white paper on the iPAQ's battery [Vin01] it is expected that the battery life will be approximately 210-225 minutes.

3.1.2 Approach. The WLAN is based on the IEEE 802.11b wireless protocol and uses Microsoft's Windows 2000 and Pocket PC 2002 operating systems with Check Point's VPN-1 SecureClient [Ltd00] software (cf., Appendix B for VPN setup information).

The hardware level security available in current wireless network cards, usually encryption algorithms such as Wired Equivalent Privacy (WEP) and RC4 (Ron's Code or Rivest's Cipher)[Lab00], is not adequate for the intended use of this system, so any card security features

are disabled. Most of these encryption algorithms have been shown to be “weak,” but even the “strong” ones only encrypt the data and leave valuable network information in the clear and thus create vulnerabilities via wireless access points. Because of these drawbacks, hardware security is not relied upon and is not used or tested.

To meet the security requirements for this system, a Virtual Private Network (VPN) was needed. Check Point, an Internet security company has developed VPN-1, a program designed for use in a Windows and Pocket PC environment [Ltd00]. This program provides the encryption, authentication, and compression features that are required.

As part of this evaluation, the effective data transfer rates of the system with the VPN incorporated is determined. The handheld battery life in real-world use is examined and some range limitations determined. The system sustains various loads depending on the range, with the effective data rates lower at larger distances.

Once the protocol and software is thoroughly examined and understood, a detailed analytical model is used to represent the system. An existing IEEE 802.11b model, as described in Chapter II, is used. Physical tests of the actual hardware are used to validate the analytical model. The transmission times and delays are monitored. These tests provide the necessary information to determine the performance of the wireless LAN.

3.2 System Boundaries

The System Under Test (SUT) includes the following (see Figure 3.1):

- i) A Dell Poweredge 4200, dual Pentium II 300 MHz CPU server with 512 MB of RAM operating Windows NT 4.1 server software and VPN-1 SecureClient 4.1 SP3 [Ltd00] server software,
- ii) Two Cisco Aironet 340 series 11 MBPS DSSS wireless network bridges [Cis00] with 50 mW output,
- iii) A 13.5 dBi Yagi directional antenna (Cisco #AIR-ANT-AKYAGI),

- iv) A 12 dBi omni-directional antenna (Cisco #AIR-ANT-AKOMNI),
- v) A Compaq Ipaq 3650 PDA [Com00] running Pocket PC 2002 with VPN-1 SecureClient 4.1 [Ltd00] client software,
- vi) A Cisco Aironet 340 series 11 MBPS DSSS wireless network PC card [Cis00] adapter,

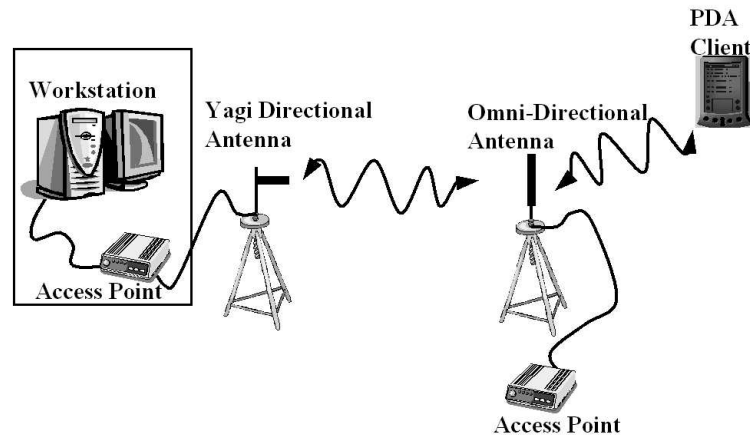


Figure 3.1 System Under Test

The Component Under Test (CUT) is the PDA, including the VPN-1 SecureClient [Ltd00] software and the wireless network card. The wireless network card adapter is a Cisco Aironet 340 series card [Cis00] which implements IEEE 802.11b. The expected operational implementation of this system is 2-5 clients that send and receive data in regular, bursty data transmissions, which means there will be few collisions. The system is tested with only one client since it is assumed clients in the operational setting will not experience excessive network contention or collisions.

3.3 System Services

This system provides bidirectional transfer of data files over a wireless link. The outcome of this service is the successful delivery of the data, a failed delivery, or a delivery of corrupted data.

3.4 Performance Metrics

The primary metric used for this research is the effective data transfer rate of the system. This rate is measured at the application layer; this provides performance information on user data transferred, which includes overhead and encapsulation. Performance at several physical distances is determined, along with the impact of various file sizes. Another metric is the PDA and jacket (with additional battery) battery life under operational use. It is important to have an accurate estimation of battery life; a short battery life means that it may be necessary to field extra battery packs along with the PDA.

3.5 Parameters

3.5.1 System. The system has several parameters of interest, they are as follows:

- i) Wireless protocol - the protocol in use affects the transmission efficiency and reliability.
- ii) Antenna distance - this range limits the maximum distances of the station from the building and the client from the station. The distance of the clients from the station affects the data transfer rates since the signal attenuates as the range increases.
- iii) Network security level - the security level affects the network performance; VPN security requires more overhead.
- iv) Level of hardware encryption - the network hardware can have WEP 128 running or not running.

3.5.2 Workload. Workload parameters that vary according to the user's needs are as follows:

- i) File size - larger files require more time to process and transmit.
- ii) Client distance from the station - the further the client is from the station, the weaker the signal and the more likely errors are to occur.

- iii) Arrival rate - excessive frequent arrivals reduce throughput because of queuing delays
- iv) Number of clients - more clients result in more transmissions which can result in collisions, reducing the throughput.

3.6 Factors

The three factors selected for this research are shown in Table 3.1:

Table 3.1 Experimental Factors

FACTOR	LEVELS
Client Distance	Close (5 ft.), Typical (700 ft.), Far (1300 ft.)
File Size	Small (10 KB), Medium (50 KB), Large (100 KB)
Network Security	Unsecure (VPN off), Secure (VPN on)

3.7 Evaluation Technique

The evaluation technique is the measurement of a hardware implementation of the system. The hardware is tested in the operational configuration. The throughput values that occur during file transfers are measured to determine the expected throughput values of the system. Values for the system with the VPN on and with the VPN off are examined separately, then compared to determine the VPN effect on the system. Using settings which replicate constant use, the PDAs (with jacket battery) battery life is tested by running the PDA until the battery has been exhausted.

3.8 Workload

Three different file sizes are offered to the system. The small file size consists of 10 KB files transmitted to the client. New files are sent only after receipt of the previous file has been acknowledged to insure independent measurement for each file. The medium file size increases the size of the packets to 50 KB. The third file size is a file of size 100 KB. These file sizes were selected to represent a range of workload conditions.

3.9 Experimental Design

3.9.1 Throughput Experiment. The throughput experiment is full factorial with 15 replications for each test. Each client distance and file size factor level combination is tested with the VPN on and off, these test cases are shown in Table 3.2. A vendor supplied software program that uses TCP and Microsoft Winsock to transfer files is used (cf., Appendix A). This software is provided by Microsoft as the standard method to transport files to and from a PDA using a network connection. The server version listens for a client connection request. When a client requests connection, the connection is made. Similar user interfaces exist on both the server and the client. The data file name can be changed and the file size selected. Once the transfer is initiated, the server starts an internal timer and begins sending the file. Once the client has received the file, it sends an acknowledgment. The server stops the timer once it receives the acknowledgment. The total transfer time is recorded as the elapsed time between server initiation and client acknowledgment. The turnaround time at the client, the elapsed time between when the client receives the file and sends the acknowledgment, as well as the transfer time for the acknowledgment message are included in the total transfer time.

Table 3.2 Throughput Test Cases

TEST	NETWORK SECURITY	CLIENT DISTANCE	FILE SIZE
1	Off	Close	Small
2	Off	Close	Medium
3	Off	Close	Large
4	Off	Typical	Small
5	Off	Typical	Medium
6	Off	Typical	Large
7	Off	Far	Small
8	Off	Far	Medium
9	Off	Far	Large
10	On	Close	Small
11	On	Close	Medium
12	On	Close	Large
13	On	Typical	Small
14	On	Typical	Medium
15	On	Typical	Large
16	On	Far	Small
17	On	Far	Medium
18	On	Far	Large

3.9.2 Battery Life Experiment. Battery life is determined by using the settings shown in Table 3.3. These settings are selected to represent constant use. Two PDAs and two PDA jackets are used. The PDA jackets contain a battery and are required for the PDA to be able to interface with the wireless network card. Since the jacket is required, the battery life of the PDA and jacket cannot be measured independently. The test consists of establishing the wireless connection with a fully charged PDA that is attached to an external power source. A 50 KB file is transmitted continuously, beginning immediately after disconnection from the external power source. The server logs the time of every file sent to and received by the client. The difference in the time of the first file transfer and the last file transfer determines the battery life of the PDA/jacket battery combination. Both PDAs are tested with both jackets; tests are replicated five times for each combination.

Table 3.3 Battery Life Test Settings

SETTING NAME	SETTING VALUE
Backlight	Always On
Battery Saver	Off
File Transfer	Continuous
File Size	50 KB

3.10 Summary

In order to determine the performance of a VPN secure WLAN with PDA clients, the system is analyzed and defined. A WLAN consisting of a PDA client, antennas, bridge, and server is established. The service provided and system parameters are examined to select the appropriate test factors. The factors selected for variation are client distance, file size, and VPN security off or on. A full factorial, 15 replication test set is run to evaluate the impact of the factor levels. The PDA and Jacket battery combination is also tested to find the expected operational duration of the client.

Along with the key metric of VPN impact, the impact of file size and data transmission range is determined to evaluate the performance of this wireless local area network. The testing involves

several key factors: the distance of the client from the station, file size, and the desired network security level. The battery of the PDA is evaluated to provide life duration under operational circumstances.

IV. Results

Research results are presented in this chapter. Detailed descriptions of test data are given. The information is summarized and an analysis given to provide a clear indication of the WLAN, PDA, and VPN software performance. The throughput experiment and battery life experiment are discussed separately.

4.1 Throughput Experiment

Results of the throughput experiment are given and summarized to provide the impact of three factors: client distance, file size, and VPN on or off. Since the primary goal of the research is to obtain a performance comparison with and without the VPN, the experiments on each will be conducted independently. The results are analyzed separately and then compared. The next two sections provide the results and factor level interactions and effects for the VPN off cases, then for VPN on. To determine the factor interactions, a full factorial design with replications is used [Jai91]. Each section begins with observed data and demonstrates how the factor level interactions and effects are determined. A summary of the factor level effects is given in a percentage form; the factor's impact is shown as the ratio of the effect and the mean throughput value. The following section compares the factor level interactions and effects of the system with and without the VPN. In the data analysis, examples are given to demonstrate how the values in the tables are generated. For a more detailed description of the process, refer to [Jai91].

4.1.1 VPN off. The results for test cases 1-9, VPN off, are discussed in the next section. The following two sections detail the effects and impact of each factor level. A summary provides a condensed table that includes the impact from each factor level.

4.1.1.1 Results. In Table 4.1, the mean of the throughput values (bps), standard deviation, coefficient of variation (C.O.V.), and a 90% confidence interval for the mean are given

for 15 replications of each test. The throughput mean values represent average throughput values observed for all 15 test replications. The standard deviation and C.O.V. describe the spread of the observed values around the mean. The throughput is determined by dividing the file size (which varies according to the file size) by the recorded time of the transfer. The standard deviation is calculated by taking the square root of the following: the number of replications times the sum of each observed throughput squared minus the square of the sum of all the observed throughput all divided by the number of replications times the number of replications minus one. The C.O.V. is determined by dividing the standard deviation by the mean. The confidence interval is found by adding and subtracting the following value from the mean: the appropriate value for the area under the standard normal curve times the standard deviation divided by the square root of the number of replications.

Table 4.1 VPN Off Results

Test	MEAN bps	STANDARD DEV. bps	C.O.V.	90% CONFIDENCE INTERVAL bps
1	12,138	127	0.010	(12,084, 12,192)
2	11,750	62	0.005	(11,724, 11,776)
3	11,674	63	0.005	(11,648, 11,701)
4	12,028	134	0.011	(11,972, 12,085)
5	11,767	54	0.005	(11,745, 11,790)
6	11,605	59	0.005	(11,580, 11,630)
7	11,976	128	0.011	(11,922, 12,031)
8	11,752	63	0.006	(11,725, 11,779)
9	11,639	62	0.006	(11,613, 11,665)

To determine the factor level interactions and effects for the client distance and file size, a Computation of Effects table is constructed (e.g., Table 4.2). This table shows the mean values for all the replications in each test. The mean throughput value for each client distance and file size is determined by row and column. The overall mean throughput of the entire data set is also determined. The effect of each client distance and file size is determined by subtracting the overall throughput from each client distance and file size throughput. For instance, the throughput mean values for the close tests (12,138, 11,750, 11,674 bps) is 11,854 bps. This value is 39 bps more than the overall mean throughput value of 11,815 bps.

Table 4.2 VPN Off Computation of Effects

all values bps	CLOSE	TYPICAL	FAR	ROW MEAN	ROW EFFECT
SMALL	12,138	12,028	11,976	12,048	233
MEDIUM	11,750	11,767	11,752	11,757	-58
LARGE	11,674	11,605	11,639	11,640	-175
COLUMN MEAN	11,854	11,801	11,790	11,815	
COLUMN EFFECT	39	-14	-25		

Table 4.3 VPN Off Interactions

all values bps	CLOSE	TYPICAL	FAR
SMALL	51	-5	-46
MEDIUM	-46	25	21
LARGE	-5	-20	25

The computation of effects shows the overall mean throughput is 11,815 bps. The effects from the client distance is small and while the effects from the file size are greater, they are still small compared to the mean throughput.

The next step is to create the Interaction Table (e.g., Table 4.3). This table is created by subtracting the overall mean, the row effect, and column effect from the throughput mean for that cell. For example, the throughput value of close and small test is 12,138 bps, subtract from this the overall mean (11815 bps), the row effect (233 bps), and the column effect (39 bps) to get the interaction value (51 bps).

The interaction effects of the factors are small, all less than 51 bps. This shows that the effects of the combined factor levels is minor compared to the overall mean throughput. The combination of factor levels creates little effect in addition to the effects caused independently from the factor levels.

The third step to the develop an Analysis Of Variance (ANOVA) table, which is Table 4.4. This table contains the following values for each component of interest: Sum of Squares (SoS), Percentage of Variation (PoV), Degrees of Freedom (DoF), Mean Square (MS), the F-Computed (F-C), and the F-Table (F-T). The SoS represents the sum of all the squared values of that component, appropriately multiplied. The PoV shows the percentage in variation that the component

has created. The DoF is the number of values that can be independently chosen for that component. The MS is the component's SoS divided by its DoF. F-C is calculated using the MS of that component divided by the MS of the errors. F-T is determined by using a statistical table. The F- values are used to demonstrate whether or not the variations created by the component are statistically significant. Take the file size component for example, the SoS is the sum of all the file size effects squared ($233^2 + (-58)^2 + (-175)^2$) multiplied by the number of replications (15) and number of levels of client distance (3), which equals 3,980,559. The PoV is 100 times its SoS (3,980,559) divided by the result of the SoS of the all the responses (18,849,011,437) minus the SoS of the mean response (18,843,773,779), which is 76%. The file size component has two degrees of freedom. The first choice of file size is independent, either small, medium, or large can be selected. The second choice can be freely selected, either of the remaining choices are available. Once the second selection is made, the final selection has no independence, it must be the last remaining file size. Because two selections can be made freely, this component has two degrees of freedom. The MS for file size is the SoS for file size (3,980,559) divided by its degrees of freedom (2), which equals 1,990,279. The F-C for file size its MS (1,990,279) divided by the MS for errors (8,039), which is 248. This is higher than the F-T value which is approximately 2.4. The final portion of this table is the standard deviation of errors, noted on the last line.

Table 4.4 VPN Off ANOVA

COMPONENT	SoS	PoV	DoF	MS	F-C	F-T
ALL MEANS	18,849,011,437		135			
MEAN RESPONSE	18,843,773,779		1			
ALL MEANS - MEAN RESPONSE	5,237,658	100	134			
CLIENT DISTANCE	109,409	2	2	54,707	7	≈2.4
FILE SIZE	3,980,559	76	2	1990,,279	248	≈2.4
INTERACTIONS	134,821	3	4	3,3705	4	≈2.1
ERRORS	1,012,869	19	126	8,039		
Standard Deviation for Errors = 89.66						

The percentage of variation (PoV) for the various components is important in the analysis of the data. For the VPN off tests, the greatest cause of variation in the mean throughput is

the file size (76%), the effect of experimental errors is a distant second (19%), the third largest percentage comes from interactions (3%), the final factor is client distance (2%). This information indicates that changing the file size is going to have the greatest impact on the throughput, but it is important to remember that this impact is small compared to the mean throughput (see the Computation of Effects table). All the F-C values are greater than the F-T values, which indicates that the effect from components are statistically significant.

The next table, Table 4.5, shows the actual effects of the factor levels and the 90% confidence intervals associated with them. The mean effect is drawn from the appropriate column or row effect of the Computation of Effects Table (Table 4.2). The standard deviation is the calculated standard deviation for that set of the sample. The confidence interval is generated by adding or subtracting 1.645 times the standard deviation from the Mean Effect. Since the error degrees of freedom is greater than 30 (actually 126), the 0.95 quantile of unit normal $Z_{0.95}$ is used instead of the t -variate. This value is 1.645. For instance, the mean response's mean effect is 11815 bps. The standard deviation is calculated by multiplying the standard deviation of the errors (89.66) times the square root of the inverse of the total degrees of freedom (1/135), which is 7.72. This is multiplied by 1.645 and subtracted from the mean effect (11,815) for the lower bound of confidence interval, 11,802. The upper bound of the confidence interval is calculated by adding the standard deviation times 1.645 to the mean effect, equaling 11,827.

Table 4.5 VPN Off Confidence Intervals for Effects

all values bps	MEAN EFFECT	STANDARD DEVIATION	CONFIDENCE INTERVAL
MEAN	11,815	7.72	(11,802, 11,827)
CLOSE	40	10.91	(22, 58)
TYPICAL	-14	10.91	(-32, 4)
FAR	-25	10.91	(-43, -7)
SMALL	233	10.91	(215, 251)
MEDIUM	-58	10.91	(-76, -40)
LARGE	-175	10.91	(-193, -157)

The standard deviation of the overall mean throughput and factor levels are all below 11 bps, much smaller than the overall mean throughput. This shows up in the confidence intervals, which are very narrow. The calculated value for the overall mean throughput and mean effects for the factor levels have a high certainty of accuracy. The confidence interval for the effect of the client distance factor level of close encompasses zero, indicating that its impact is not significantly different from a zero effect.

The final table, Table 4.6, addresses the 90% confidence intervals for the interactions. For each client distance and file size combination, the confidence interval for effect on the overall mean throughput is given. The standard deviation for the interactions is given on the bottom of the table. This was calculated by multiplying the standard deviation of the errors (89.66) by the square root of the result of degrees of freedom in the file size (2) multiplied by the degrees of freedom in the client distances (2) divided by the total degrees of freedom (135) to get 13.37. These confidence intervals were generated in the same fashion as the confidence intervals for effects.

Table 4.6 VPN Off Confidence Intervals for Interactions

all values bps	CLOSE	TYPICAL	FAR
SMALL	(29, 73)	(-27, 17)	(-68, -24)
MEDIUM	(-68, -24)	(3, 47)	(-1, 43)
LARGE	(-27, 17)	(-42, 2)	(3, 47)
Standard Deviation for Interactions = 13.37			

It is important to recognize that two of the three confidence intervals for the typical client distance include zero, demonstrating that they are not significantly different from a zero effect.

4.1.1.2 Client Distance Impact. Table 4.7 summarizes the factor level impact of the client distance. The ratio of the effect to the mean throughput is less than 0.5% for each factor level. This impact is not large enough to support a claim of significant throughput impact due to changing the client distance.

Table 4.7 VPN Off Client Distance Impact

	CLOSE	TYPICAL	FAR
EFFECT	39 ± 18 bps	-14 ± 18 bps	-25 ± 18 bps
IMPACT	0.33%	-0.12%	-0.21%
Mean Throughput = 11,815 bps			

4.1.1.3 *File Size Impact.* Table 4.8 summarizes the factor level impact of the file size. The level of the file size factor has minor impact on the throughput of the network. The ratio of the effect to the mean throughput is almost 2% for both the small and large factor levels. This impact demonstrates that a larger file size will decrease throughput, but not drastically.

Table 4.8 VPN Off File Size Impact

	SMALL	MEDIUM	LARGE
EFFECT	233 ± 18 bps	-58 ± 18 bps	-175 ± 18 bps
IMPACT	1.97%	-0.49%	-1.48%
Mean Throughput = 11,815 bps			

4.1.1.4 *Summary.* Table 4.9 summarizes the factor level impact for client distance and file size. This table shows the impact caused by each factor relative to the mean throughput. The impact of the client distance and file size factors are minor. While the impact due to changing the file size factor is most significant, it's impact only results in a change of about 2% in the mean throughput.

Table 4.9 VPN Off Factor Level Impact

	CLIENT DISTANCE			FILE SIZE		
	CLOSE	TYPICAL	FAR	SMALL	MEDIUM	LARGE
IMPACT	0.33%	-0.12%	-0.21%	1.97%	-0.49%	-1.48%
Mean Throughput = 11,815 bps						

4.1.2 *VPN On.* The results for test cases 10-18, VPN on, are discussed in the next section. The following two sections detail the effects and impact of each factor level. A summary provides a condensed table that includes the impact from each factor level. Examples are given in the previous section for each of the tables included below, they will not be repeated in this section.

4.1.2.1 *Results.* Table 4.10 shows the same information, the Results, for the VPN

on tests as Table 4.1 for VPN off.

Table 4.10 VPN On Results

Test	MEAN bps	STANDARD DEV. bps	C.O.V.	90% CONFIDENCE INTERVAL bps
10	11,749	40	0.003	(11,732, 11,766)
11	11,438	39	0.003	(11,421, 11,454)
12	11,226	36	0.003	(11,211, 11,241)
13	11,645	71	0.006	(11,615, 11,675)
14	11,348	37	0.003	(11,332, 11,364)
15	11,192	29	0.003	(11,180, 11,205)
16	11,670	64	0.005	(11,643, 11,697)
17	11,279	49	0.004	(11,258, 11,300)
18	10,972	22	0.002	(10,962, 10,981)

Table 4.11 shows the same information, Computation of Effects, for the VPN on tests as Table 4.2 for VPN off. The computation of effects shows the overall mean throughput is 11,391 bps. The effects from the client distance is small and while the effects from the file size are greater, they are still small compared to the mean throughput.

Table 4.12 shows the same information, the Interactions, for the VPN on tests as Table 4.3 for VPN off. The interaction effects of the factors are small, all less than 70 bps. This shows that the effects of the combined factor levels is minor compared to the overall mean throughput. The combination of factor levels creates little effect in addition to the effects caused independently from the factor levels.

Table 4.13 shows the same information, the ANOVA, for the VPN on tests as Table 4.4 for VPN off. The percentage of variation (PoV) for the various components is important in the analysis of the data. For the VPN on tests, the greatest cause of variation in the mean throughput is the file

Table 4.11 VPN On Computation of Effects

all values bps	CLOSE	TYPICAL	FAR	ROW MEAN	ROW EFFECT
SMALL	11,749	11,645	11,670	11,688	297
MEDIUM	11,437	11,348	11,279	11,355	-36
LARGE	11,226	11,192	10,972	11,130	-261
COLUMN MEAN	11,471	11,395	11,307	11,391	
COLUMN EFFECT	80	4	-84		

Table 4.12 VPN On Interactions

all values bps	CLOSE	TYPICAL	FAR
SMALL	-19	-47	66
MEDIUM	3	-11	8
LARGE	16	58	-74

Table 4.13 VPN On ANOVA

COMPONENT	SoS	PoV	DoF	MS	F-C	F-T
ALL MEANS	17,524,889,797		135			
MEAN RESPONSE	17,516,681,622		1			
ALL MEANS - MEAN RESPONSE	8,208,175	100	134			
CLIENT DISTANCE	606,555	7	2	303,278	147	≈2.4
FILE SIZE	7,095,747	87	2	3,547,874	1,720	≈2.4
INTERACTIONS	245,940	3	4	61,485	30	≈2.1
ERRORS	259,932	3	126	2,063		
Standard Deviation for Errors = 45.42						

size (87%), the effect of client distance is a distant second (7%), the third largest percentage comes from interactions (3%), the final factor is experimental errors (3%) . This information indicates that changing the file size is going to have the greatest impact on the throughput, but it is important to remember that this impact is small compared to the mean throughput (see the Computation of Effects table). All the F-C values are greater than the F-T values, which indicates that the effect from components are statistically significant.

Table 4.14 VPN On Confidence Intervals for Effects

all values bps	MEAN EFFECT	STANDARD DEVIATION	CONFIDENCE INTERVAL
MEAN	11391	3.91	(11385, 11397)
CLOSE	80	5.528	(71, 89)
TYPICAL	4	5.53	(-5, 13)
FAR	-84	5.53	(-93, -75)
SMALL	297	5.528	(288, 306)
MEDIUM	-36	5.53	(-45, -27)
LARGE	-261	5.528	(-270, -252)

Table 4.14 shows the same information, Confidence Intervals for Effects, for the VPN on tests as Table 4.5 for VPN off. The standard deviation of the overall mean throughput and factor levels are all below 6 bps, much smaller than the overall mean throughput. This shows up in the

confidence intervals, which are very narrow. The calculated value for the overall mean throughput and mean effects for the factor levels have a high certainty of accuracy. The confidence interval for the effect of the client distance factor level of close encompasses zero, indicating that its impact is not significantly different from a zero effect.

Table 4.15 shows the same information, Confidence Intervals for Interactions, for the VPN on tests as Table 4.6 for VPN off.

Table 4.15 VPN On Confidence Intervals for Interactions

all values bps	CLOSE	TYPICAL	FAR
SMALL	(-30, -8)	(-59, -36)	(-55, -77)
MEDIUM	(-8, 14)	(-22, 0)	(-3, 19)
LARGE	(5, 27)	(47, 70)	(-85, -63)
Standard Deviation for Interactions = 6.77			

4.1.2.2 *Client Distance Impact.* Table 4.16 summarizes the factor level impact of the client distance. The ratio of the effect to the mean throughput is less than 0.8% for each factor level. This impact is not large enough to support a claim of significant throughput impact due to changing the client distance.

Table 4.16 VPN On Client Distance Impact

	CLOSE	TYPICAL	FAR
EFFECT	80 ± 9 bps	4 ± 9 bps	-84 ± 9 bps
IMPACT	0.70%	0.04%	-0.74%
Mean Throughput = 11,391 bps			

4.1.2.3 *File Size Impact.* Table 4.17 summarizes the factor level impact of the File Size. The level of the file size factor has minor impact on the throughput of the network. The ratio of the effect to the mean throughput is about 2.5% for both the small and large factor levels. This impact demonstrates that a higher file size will decrease throughput, but not drastically.

4.1.2.4 *Summary.* Table 4.18 summarizes the factor level impact for client distance and File Size. This table shows the impact caused by each factor relative to the mean throughput. The impact of the client distance and file size factors are minor. While the impact do to changing

Table 4.17 VPN On File Size Impact

	SMALL	MEDIUM	LARGE
EFFECT	297 ± 9 bps	-36 ± 9 bps	-261 ± 9 bps
IMPACT	2.61%	-0.32%	-2.29%
Mean Throughput = 11,391 bps			

the file size factor is most significant, it's impact only results in a change of about 2.5% in the mean throughput.

Table 4.18 VPN On Factor Level Impact

	CLIENT		DISTANCE	FILE		SIZE
	CLOSE	TYPICAL	FAR	SMALL	MEDIUM	LARGE
IMPACT	0.70%	0.04%	-0.74%	2.60%	-0.32%	-2.29%
Mean Throughput = 11391 bps						

4.1.3 Comparison. This section compares the mean throughput values and factor level effects of the VPN off and VPN on cases analyzed above. This provides a side-by-side comparison of the network with and without the VPN installed.

4.1.3.1 Throughput. Table 4.19 is a comparison of mean throughput values for each of the test cases between the VPN off and VPN on. The final line of the table shows the percent decrease from the VPN off throughput to the VPN on throughput. The decrease ranges from 2.55% to 5.75%. This small percentage decrease is not operationally significant for most WLAN implementations. The performance decrease is similar to the decrease in [Cus01]. This report stated a 7% throughput decrease in an 802.11 wireless network with a Windows 2000 laptop. Figure 4.1 shows the trend of this decrease. As the file size and client distance increase, the decrease due to the VPN rises. The VPN requires more acknowledgment, control, and authentication messages. Also the encryption algorithm increases overhead. These factors make the VPN implementation more sensitive to factor changes.

4.1.3.2 Client Distance Impact. Table 4.20 provides a comparison of the client distance impact determined for both the VPN off and VPN on. Figure 4.2 shows graphs of the

Table 4.19 Mean Throughput Comparison

all values bps	MEAN	CLOSE SMALL	CLOSE MEDIUM	CLOSE LARGE
VPN OFF	11,815	12,138	11,750	11,674
VPN ON	11,391	11,749	11,437	11,226
Decrease	3.59%	3.20%	2.66%	3.84%

	TYPICAL SMALL	TYPICAL MEDIUM	TYPICAL LARGE	FAR SMALL	FAR MEDIUM	FAR LARGE
VPN OFF	12,028	11,767	11,605	11,976	11,752	11,639
VPN On	11,645	11,348	11,192	11,670	11,279	10,972
Decrease	3.19%	3.56%	3.55%	2.56%	4.03%	5.74%

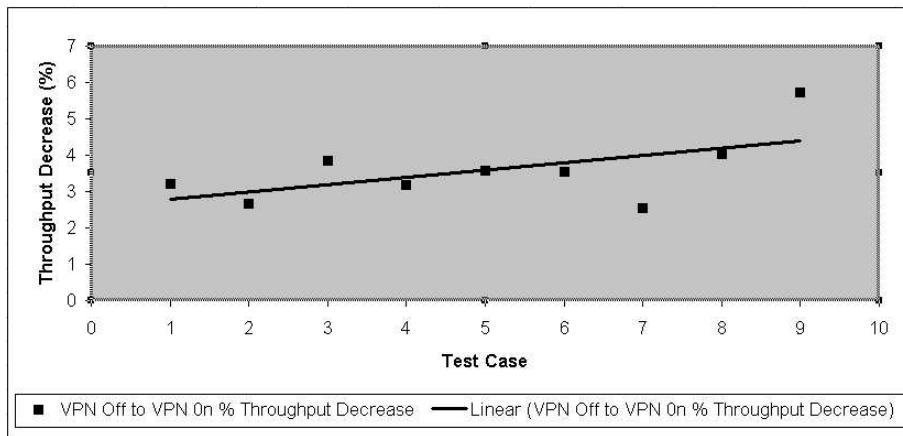


Figure 4.1 VPN On Throughput Decrease

impacts. The ranges tested did not stress the system enough. The client distance impact for both VPN off and on are small, with the VPN on impact slightly larger. This shows that the throughput of the VPN on cases is more vulnerable to changes. The decrease in performance caused by the greater ranges is compounded when the VPN is running due to increased messaging requirements and overhead. Since this impact is insignificant, further distances could be supported without notably decreasing throughput. The use of the omni-directional antenna to communicate with the client and with the Yagi directional antenna is quite effective. The omni-directional antenna provides client use flexibility while supporting reachback to the Yagi antenna and wired network.

4.1.3.3 *File Size Impact.* Table 4.21 provides a comparison of the file size impact determined for both the VPN off and VPN on. Figure 4.3 shows graphs of the impacts. The file

Table 4.20 Client Distance Impact Comparison

	CLOSE	TYPICAL	FAR
VPN OFF	0.33%	-0.12%	-0.21%
VPN ON	0.70%	0.04%	-0.74%

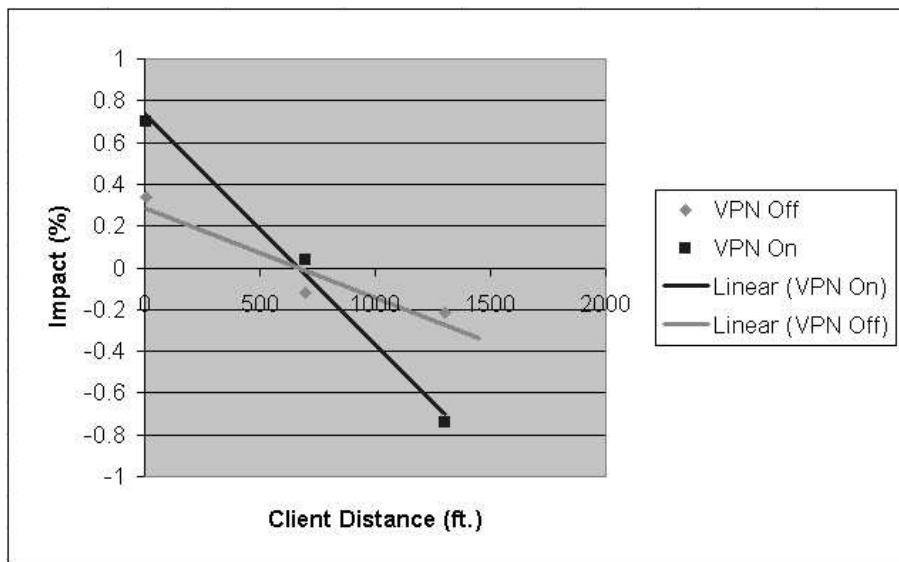


Figure 4.2 Client Distance Impact Graph

size impact for both VPN off and on are small, with the VPN on impact slightly larger. This shows that the throughput of the VPN on cases is more vulnerable to changes in the same manner as the client distances. The impact from increasing the file size is linear. By following the data trend, a 150 KB file would decrease the throughput by 4% to 5% and a 300 KB file by 10% to 13%.

Table 4.21 File Size Impact Comparison

	SMALL	MEDIUM	LARGE
VPN OFF	1.97%	-0.49%	-1.48%
VPN ON	2.61%	-0.32%	-2.29%

4.1.3.4 *Summary.* The test data supports the hypothesis, that the VPN would negatively effect the throughput, but not to unacceptable levels. The interesting fact about the data is the low throughput achieved. The wireless card used is rated at 11 Mbps (11,000,000 bps) but the network's highest measured data rate was 12,290 bps. Using Bianchi's model [Bia00] the maximum possible throughput is 9,130,00 bps ($0.83 * 11 \text{ Mbps}$), two orders of magnitude better

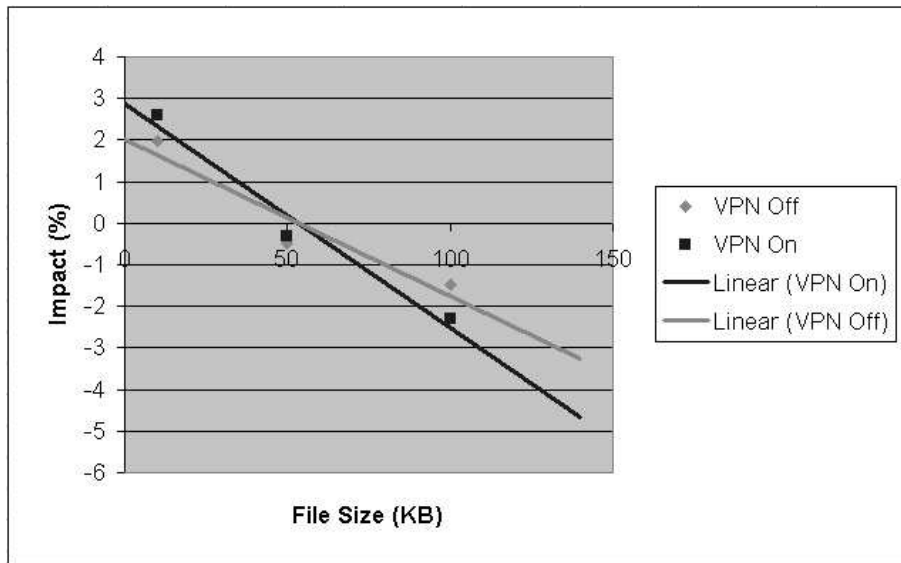


Figure 4.3 File Size Impact Graph

performance than the tested system. The transfer program was examined to understand its impact on throughput. This is the Microsoft supplied program for transferring files onto Pocket PCs via Winsock. While the code was changed slightly, the changes only affected the visual GUI of the program and the data recording. These changes do not affect the throughput. The data recording is done only on the server size, after receipt of the acknowledgement from the client and before another file is sent. In order to compare the performance of the Pocket PC O/S to the Windows 200 O/S, the program is compiled using Visual Basic Studio, as opposed to the native Embedded Visual Basic Studio. System call names and other O/S environment information was changed. Again, these changes should not affect the throughput. When running a Windows 2000 laptop, the transfer program achieved data rates up to 2.22 Mbps, much closer to Bianchi's maximum (cf., Appendix D). Testing conducted by the University Corporation for Atmospheric Research on wireless networks (without PDAs) show laptop throughput values of up to 5 Mbps with VPN implementation decreasing that value by 7% [Cus01]. This information indicates that while the transfer program could be more efficient, it is not responsible for the low throughput values on the PDA. It should be noted that the laptop had the same network hardware configured with the

same setup as the PDA. Investigation of the Pocket PC O/S through Microsoft's website, points to information on its limitations. This O/S is an extremely scaled-down version of the Windows operating system. The system calls are not as varied or usable as in Windows. The Winsock controls are particularly coarse and inefficient. User created Winsock controls are suggested as a solution in [Vic02], stating a 200% performance increase in his implementation. Another O/S limitation is in multi-tasking ability. The Pocket PC has no multi-tasking capability, there is no way to work on one task while another task is waiting for an event to occur. The PDA memory and processor constraints also contribute to inefficient data transfer.

4.2 Battery Life Experiment Results

This section provides the results of the battery life duration testing. Table 4.22 gives a summary of the data results. The combinations represent each PDA and jacket combination tested. The mean is the average amount of time the PDA and jacket combination lasted. The standard deviation and coefficient of variance (C.O.V.) describe the spread of the observed values around the mean. A 90% confidence interval of the mean is also given. The standard deviation is calculated by taking the square root of the following: the number of replications times the sum of each observed throughput squared minus the square of the sum of all the observed throughput all divided by the number of replications times the number of replications minus one.

Table 4.22 Battery Life duration Results

COMBINATION	MEAN min	STANDARD DEVIATION min	C.O.V.	90% CONFIDENCE INTERVAL min
1	163.70	11.75	0.0718	(155.05, 172.34)
2	145.11	2.51	0.0173	(143.26, 146.96)
3	179.83	3.81	0.0212	(177.02, 182.64)
4	169.10	4.19	0.0248	(166.02, 172.18)
Overall	164.44	14.27	0.0868	(159.19, 169.68)

The overall mean duration of the PDA and jacket combination is 164.44 minutes. With a 90% confidence, any given combination should last from 159.19 to 169.68 minutes. Examining the Constant Use Battery Life Testing tables in [Vin01], it is apparent that the jacket battery provides

50-66% of the PDA and jacket combination's duration. The shorter the duration, the higher the percentage that the jacket contributes. Since the observed mean life duration is shorter than all cases provided in [Vin01], it is safe to assume that the jacket provides at least 60% of the duration in this testing. Extra jackets will provide about 98.6 additional minutes of power.

4.3 Summary

This chapter provides the data observed from hardware testing. Data is analyzed and summarized to provide useful information on the performance of the system. The throughput effects of the various factor levels are determined and presented as a ratio to the overall mean throughput. The system is analyzed independently with the VPN off and on, those cases are compared, and a combined analysis is presented. The data shows a mean throughput value of 11,815 bps for the VPN off tests and 11,391 bps for the VPN on tests. These values are much lower than anticipated, but the VPN effect is small. The client distance impact is nearly negligible, while file size had a small impact on performance. The battery life duration results are given and analyzed, the PDA/jacket combination has a duration of around 164 minutes.

V. Conclusions

The research significance is explained in this chapter. Results are summarized and a conclusion is given. Some implications and research impact is provided along with recommendations for possible topics of further research in this area.

5.1 Results

Results detailed in Chapter IV are given here in concise summarized form. The throughput experiment and battery life experiment are discussed separately.

5.1.1 Throughput Experiment. Results of the throughput experiment are summarized in Tables 5.1 and 5.2. To demonstrate the impact of the factor levels, the percentage of the effect of each factor level to the overall mean throughput is given. Per Table 5.1, the mean throughput of the VPN off tests is 11,815 bps and 11,391 bps for the VPN on tests, representing a 3.59% decrease in performance when the VPN is on. Per Table 5.2, the impact of the file size and client distance is small, $\leq 3\%$. The impact from varying factor levels is greater in the VPN on tests than the VPN off, due to higher overhead and messaging requirements.

5.1.2 Battery Life Experiment. The battery life of the PDA and jacket battery combination (two PDAs and two jackets) is observed to be about 164 minutes on average, with additional

Table 5.1 Throughput Comparison

Test Cases:	Mean	1, 10	2, 11	3, 12	4, 13
VPN Off	11,815	12,138	11,750	11,674	12,028
VPN On	11,391	11,749	11,438	11,226	11,645
Percent Decrease	3.59%	3.21%	2.66%	3.84%	3.19%
Test Cases:	5, 4	6, 15	7, 16	8, 17	9, 18
VPN Off	11,767	11,605	11,976	11,752	11,639
VPN On	11,348	11,192	11,670	11,279	10,972
Percent Decrease	3.56%	3.55%	2.55%	4.03%	5.74%

Table 5.2 Factor Impact

	CLIENT DISTANCE			FILE SIZE		
	CLOSE	TYPICAL	FAR	SMALL	MEDIUM	LARGE
VPN OFF	0.33%	-0.12%	-0.21%	1.97%	-0.49%	-1.48%
VPN ON	0.70%	0.04%	-0.74%	2.61%	-0.32%	-2.29%

jackets adding about 99 minutes each. The tested conditions represent constant network traffic over the wireless link. Less excessive use of the wireless link would result in longer battery life.

5.2 Conclusion

Research results indicate very poor performance of a Wireless Local Area Network utilizing PDAs. Of the three factors (client distance, file size, and VPN), the throughput is effected most by the VPN implementation and slightly by increased file size. The client distance factor has virtually no effect on throughput. The impact of each of these factor levels is small when compared to the magnitude of the overall mean throughput ($\leq 6\%$). The average throughput of the network with the PDA client is much lower than expected, $\approx 11,500$ bps versus 9,130,00 bps provided by the analytical model [Bia00]. This is attributed to several factors, with degradation primarily resulting from limitations of the PDA hardware and O/S. Because of the low throughput values achieved (regardless if VPN is off or on), an operational WLAN with PDAs (as tested) is not feasible. Operational use of the network tested would require an in-depth analysis of the type of network traffic and performance required to maintain functionality. To deploy such a system, custom designed Winsock controls would need to be implemented to minimize limitations imposed by the PDA. As PDA technology continues to develop, future hardware and O/S functionality may provide a more robust platform for network communications. The battery life duration was a little shorter than anticipated. This is likely due to the fact that the wireless network card adapter requires more power than the average PC card tested by Compaq [Vin01].

Research into the Winsock implementation and network communications for the Pocket PC operating system could prove very beneficial. It is expected that a more optimized communication

method can improve the throughput drastically. A Windows 2000 implementation of the client software (cf., Appendix D) with more robust Winsock controls improved the throughput by two orders-of-magnitude. Another interesting area of study would be in environmental effects. Terrain, buildings, weather, etc. impact analysis would provide a comprehensive understanding of the PDAs capabilities in various situations. Because of an observed lack of impact due to client distance in this research, further studies into range limitations should be conducted.

5.3 Summary

The effect of the VPN implementation on this network is small, but significant. The effect from the file size and client distance are smaller than the effect from the VPN, in fact the effect from client distance was almost negligible. Overall, the PDA client's throughput rates, ≈ 11500 bps, are not comparable to those of a laptop computer, 2,220,000 bps. The PDA operating system and environment proved to be the limiting factor in the performance of this wireless local area network. The life duration of the PDA and jacket combination was determined to be about 164 minutes, with additional jackets adding about 99 more minutes.

Appendix A. Transfer Program Code

A.1 Introduction

The basic transfer program used in testing was obtained from the Microsoft website at: <http://support.microsoft.com/directory/article.asp?ID=KB;EN-US;Q275004>. This program was designed to transfer bitmap files to a Pocket PC PDA via Winsocket. The program was slightly modified to incorporate requirements for this research, such as custom GUI and data logging.

A.2 Server Side Code

The code for the server side version of the transfer program is included below. The code is in Visual Basic and was edited using Visual Basic Studio 6.0. The file `server.vbp` is the Visual Basic Project file which basically provides header information. The file `server.frm` is the Visual Basic Form file which contains all the methods, control functions, and variables.

A.2.1 server.vbp.

```
Type=Exe
Form=server.frm
Reference=*\G{00020430-0000-0000-C000-000000000046}#2.0#0#
C:\WINNT\System32\stdole2.tlb#OLE Automation
Object={248DD890-BB45-11CF-9ABC-0080C7E7B78D}#1.0#0; MSWINSCK.OCX
IconForm="Form1"
Startup="Form1"
ExeName32="server.exe"
Command32=""
Name="Project1"
HelpContextID="0"
```

CompatibleMode="0"
MajorVer=1
MinorVer=0
RevisionVer=0
AutoIncrementVer=0
ServerSupportFiles=0
VersionCompanyName="wpafb"
CompilationType=0
OptimizationType=0
FavorPentiumPro(tm)=0
CodeViewDebugInfo=0
NoAliasing=0
BoundsCheck=0
OverflowCheck=0
FlPointCheck=0
FDIVCheck=0
UnroundedFP=0
StartMode=0
Unattended=0
Retained=0
ThreadPerObject=0
MaxNumberOfThreads=1

[MS Transaction Server]

AutoRefresh=1

A.2.2 server.frm.

VERSION 5.00

Object = "{248DD890-BB45-11CF-9ABC-0080C7E7B78D}#1.0#0"; "MSWINSCK.OCX"

Begin VB.Form Form1

 Caption = "Server"

 ClientHeight = 6375

 ClientLeft = 60

 ClientTop = 345

 ClientWidth = 5520

 LinkTopic = "Form1"

 ScaleHeight = 6375

 ScaleWidth = 5520

 StartPosition = 3 'Windows Default

Begin MSWinsockLib.Winsock Winsock3

 Left = 2880

 Top = 4920

 _ExtentX = 741

 _ExtentY = 741

 _Version = 393216

End

Begin VB.Timer Timer2

 Left = 2880

 Top = 2760

End

Begin VB.TextBox Text1

```
    Height      = 375
    Left        = 120
    TabIndex    = 15
    Text        = "Text1"
    Top         = 5400
    Width       = 5175
End
Begin VB.FileListBox File1
    Height      = 1065
    Left        = 120
    TabIndex    = 14
    Top         = 4320
    Width       = 2175
End
Begin VB.DirListBox Dir1
    Height      = 990
    Left        = 120
    TabIndex    = 13
    Top         = 3360
    Width       = 2175
End
Begin VB.DriveListBox Drive1
    Height      = 315
    Left        = 120
    TabIndex    = 12
    Top         = 3000
```

```

        Width          = 2175
End
Begin VB.CommandButton Command7
    Caption          = "stop cont"
    Height          = 375
    Left            = 1440
    TabIndex        = 8
    Top             = 2400
    Width           = 1095
End
Begin VB.CommandButton Command6
    Caption          = "send lg cont"
    Height          = 375
    Left            = 1440
    TabIndex        = 7
    Top             = 1680
    Width           = 1095
End
Begin VB.CommandButton Command5
    Caption          = "send sm cont"
    Height          = 375
    Left            = 1440
    TabIndex        = 6
    Top             = 960
    Width           = 1095
End

```

```

Begin VB.CommandButton Command4

    Caption        =   "send large"

    Height         =   375

    Left           =   120

    TabIndex       =   5

    Top            =   1680

    Width          =   1095

End

Begin VB.Timer Timer1

    Left           =   2880

    Top            =   3240

End

Begin MSWinsockLib.Winsock Winsock2

    Left           =   2880

    Top            =   4440

    _ExtentX       =   741

    _ExtentY       =   741

    _Version       =   393216

End

Begin MSWinsockLib.Winsock Winsock1

    Left           =   2880

    Top            =   3840

    _ExtentX       =   741

    _ExtentY       =   741

    _Version       =   393216

End

```

```
Begin VB.CommandButton Command3

    Caption        =   "disconnect"

    Height         =   375

    Left           =   120

    TabIndex       =   2

    Top            =   2400

    Width          =   1095
```

End

```
Begin VB.CommandButton Command2

    Caption        =   "send small"

    Height         =   375

    Left           =   120

    TabIndex       =   1

    Top            =   960

    Width          =   1095
```

End

```
Begin VB.CommandButton command1

    Caption        =   "connect"

    Height         =   375

    Left           =   120

    TabIndex       =   0

    Top            =   240

    Width          =   1095
```

End

```
Begin VB.Label Label5

    Caption        =   "sum "
```

```

    Height      = 255
    Left        = 2880
    TabIndex    = 11
    Top         = 1920
    Width       = 975
End

Begin VB.Label Label4

    Caption     = "data rate"
    Height      = 255
    Left        = 2880
    TabIndex    = 10
    Top         = 960
    Width       = 975
End

Begin VB.Label Label3

    Caption     = "sum"
    Height      = 375
    Left        = 2880
    TabIndex    = 9
    Top         = 2280
    Width       = 1815
End

Begin VB.Label Label2

    Caption     = "connection status"
    Height      = 375
    Left        = 1440

```



```

        TabIndex      = 4

        Top           = 240

        Width         = 2055

    End

    Begin VB.Label Label1

        Caption        = "send rate"

        Height         = 375

        Left           = 2880

        TabIndex       = 3

        Top            = 1320

        Width          = 1695

    End

End

Attribute VB_Name = "Form1"

Attribute VB_GlobalNameSpace = False

Attribute VB_Creatable = False

Attribute VB_PredeclaredId = True

Attribute VB_Exposed = False

Option Explicit

Private Const WS_VERSION_REQD = &H101

Private Const WS_VERSION_MAJOR = WS_VERSION_REQD \ &H100 And &HFF&

Private Const WS_VERSION_MINOR = WS_VERSION_REQD And &HFF&

Private Const MIN_SOCKETS_REQD = 1

Private Const SOCKET_ERROR = -1

Private Const WSADescription_Len = 256

```

```
Private Const WSASYS_Status_Len = 128

Dim timecount As Long ' time count in 10 millisecond increments

Dim trate As Long ' computed data transfer rate

Dim bsmallData() As Byte ' small data packet to send

Dim blargeData() As Byte ' large data packet to send

Dim dataheader() As Byte

Dim ismallsize As Long ' size of small data packet to send

Dim ilargesize As Long ' size of large data packet to send

Dim isize As Long ' size of data packet for current test

Dim sdata As String ' received control information from hand held

Dim sdata2 As String

Dim isendrate As Long ' computed data transfer rate

Dim itrate2 As Long

Dim ismallsum As Long ' sum of the data s set

Dim ilargesum As Long ' sum of the data l set

Dim isum As Long

Dim iwitchtest As Integer

Dim afilename As String

Dim iwait As Integer

Dim iwait2 As Integer

Dim istop As Integer

Dim ATEMP As String

Dim ATEMP2 As String

Dim createdata As Integer

Dim idataset As Integer
```

```
Dim ndatasets As Long
Dim nospeeddata As Integer
Dim ticktemp As Long
Dim tickstart As Long
Dim tickmiddle As Long
Dim tickstop As Long
Dim tickfirstrecv As Long
Dim tickrecvdone As Long
Dim igettick As Integer
Dim iget1tick As Integer
Dim iget2tick As Integer
Dim firstack As Integer
Dim ctime As Long
```

```
Private Type HOSTENT
```

```
    hName As Long
    hAliases As Long
    hAddrType As Integer
    hLength As Integer
    hAddrList As Long
```

```
End Type
```

```
Private Type WSADATA
```

```
    wversion As Integer
    wHighVersion As Integer
    szDescription(0 To WSA_DESCRIPTION_LEN) As Byte
```

```

    szSystemStatus(0 To WSASYS_Status_Len) As Byte

    iMaxSockets As Integer

    iMaxUdpDg As Integer

    lpszVendorInfo As Long

End Type

Private Declare Function WSAGetLastError Lib "WSOCK32.DLL" () As Long

Private Declare Function GetTickCount Lib "kernel32" () As Long

Private Declare Function WSASStartup Lib "WSOCK32.DLL" ( _
    ByVal wVersionRequired As Long, _
    lpWSAData As WSADATA) As Long

Private Declare Function WSACleanup Lib "WSOCK32.DLL" () As Long

Private Declare Function gethostname Lib "WSOCK32.DLL" ( _
    ByVal hostname As String, _
    ByVal HostLen As Long) As Long

Private Declare Function gethostbyname Lib "WSOCK32.DLL" ( _
    ByVal hostname As String) As Long

Private Declare Sub RtlMoveMemory Lib "kernel32" ( _
    hpvDest As Any, _
    ByVal hpvSource As Long, _
    ByVal cbCopy As Long)

Private Function hibernate(ByVal wParam As Integer)

    hibernate = wParam \ &H100 And &HFF&

```

```
End Function
```

```
Private Function lobyte(ByVal wParam As Integer)
```

```
    lobyte = wParam And &HFF&
```

```
End Function
```

```
Private Sub Command4_Click()
```

```
Rem send large
```

```
    Close #1
```

```
    Open Text1 For Append As #1
```

```
    Text1.Enabled = False
```

```
    disable_all_buttons
```

```
    isum = ilargesum
```

```
    isize = ilargesize
```

```
    iwitchtest = 4
```

```
    iwait = 1
```

```
    igettick = 0
```

```
    iget1tick = 0 ' set iget1tick =0 so only tick at first winsock2 "10" is recorded
```

```
    If ndatasets > 1 Then blargeData(ilargesize - 1) = CByte(20)
```

```
    tickstart = GetTickCount()
```

```
    If ndatasets > 1 Then
```

```
        For idataset = 1 To ndatasets - 1
```

```
            Winsock2.SendData blargeData()
```

```
            DoEvents
```

```
        Next idataset
```

```
        blargeData(ilargesize - 1) = CByte(0)
```

```

End If

Winsock2.SendData blargeData()

Do While iwait = 1

DoEvents

Loop

Close #1

Label3.Caption = Str(isum)

End Sub

Private Sub Command5_Click()

Rem small data set send continusly

Close #1

Open Text1 For Append As #1

Text1.Enabled = False

istop = 0

Do While istop = 0

disable_all_buttons

Command7.Enabled = True ' stop send

isum = ismallsum

isize = ismallsize

iwitchtest = 5

iwait = 1

igettick = 0

iget1tick = 0 ' set iget1tick =0 so only tick at first winsock2 "10" is recorded

If ndatasets > 1 Then bsmallData(ismallsize - 1) = CByte(20)

tickstart = GetTickCount()

```

```

If ndatasets > 1 Then
    For idataset = 1 To ndatasets - 1
        Winsock2.SendData bsmallData()
        DoEvents
    Next idataset
    bsmallData(ismallsize - 1) = CByte(0)
End If
Label3.Caption = "before send" + Str(GetTickCount())
Winsock2.SendData bsmallData()
Label3.Caption = "before iwait" + Str(GetTickCount())
Do Until iwait = 0
    DoEvents
Loop
iwait2 = 1
Label3.Caption = "before iwait2" + Str(GetTickCount())
Timer2 = True
Do Until iwait2 = 0 ' wait here for 1 second or you get a key from remote
    DoEvents
Loop
Timer2 = False
Loop
Close #1
iwait2 = 1
Timer2 = True
Do Until iwait2 = 0
DoEvents

```

```

Loop

Timer2 = False

enable_all_buttons

End Sub

Private Sub Command6_Click()

Rem large data continuous send

    istop = 0

    Close #1

    Open Text1 For Append As #1

    Text1.Enabled = False

    Do While istop = 0

        disable_all_buttons

        Command7.Enabled = True ' stop send

        isum = ilargesum

        isize = ilargesize

        iwitchtest = 6

        iwait = 1

        igettick = 0

        iget1tick = 0 ' set iget1tick =0 so only tick at first winsock2 "10" is recorded

        If ndatasets > 1 Then blargeData(ilargesize - 1) = CByte(20)

        tickstart = GetTickCount()

        If ndatasets > 1 Then

            For idataset = 1 To ndatasets - 1

```



```

        Winsock2.SendData blargeData()

        DoEvents

        Next idataset

        blargeData(ilargesize - 1) = CByte(0)

    End If

    Label3.Caption = "before send" + Str(GetTickCount())

    Winsock2.SendData blargeData()

    Label3.Caption = "before iwait" + Str(GetTickCount())

    Do Until iwait = 0 ' iwait set to 0 at data arival

        DoEvents

    Loop

    iwait2 = 1

    Label3.Caption = "before iwait2" + Str(GetTickCount())

    Timer2 = True

    Do Until iwait2 = 0 ' wait here for 1 second or you get a key from remote

        DoEvents

    Loop

    Timer2 = False

Loop

Close #1

enable_all_buttons

End Sub

```

```
Private Sub Command7_Click()
```

```
Rem Stop Cont or stop test so you can do other tests
```

```
If iwitchtest <> 5 And iwitchtest <> 6 Then
```

```
command1.Enabled = False 'connect
```

```
Command2.Enabled = True ' send small
```

```
Command3.Enabled = True ' disconnect
```

```
Command4.Enabled = True ' send large
```

```
Command5.Enabled = True ' send small cont
```

```
Command6.Enabled = True ' send large cont
```

```
Command7.Enabled = False ' stop send
```

```
Text1.Enabled = True
```

```
End If
```

```
istop = 1
```

```
await = 0
```

```
End Sub
```

```
Private Sub Dir1_Change()
```

```
File1 = Dir1
```

```
If Len(Dir1) < 4 Then
```

```
Text1 = Dir1 + afilename
```

```
Else
```

```
Text1 = Dir1 + "\" + afilename
```

```
End If
```

```
End Sub
```

```
Private Sub Drive1_Change()
```

```
Dir1 = Drive1
```

```
If Len(Dir1) < 4 Then
Text1 = Dir1 + filename
Else
Text1 = Dir1 + "\" + filename
End If
End Sub
```

```
Private Sub File1_Click()
filename = File1
If Len(Dir1) < 4 Then
Text1 = Dir1 + filename
Else
Text1 = Dir1 + "\" + filename
End If
End Sub
```

```
Private Sub Form_Load()
    Dim i As Long
    Dim j As Integer
    Dim isum As Long
    Dim ibyte As Byte
    ATEMP = ""
    firstack = 0 ' firstack = 0 is for first acknowledgement of data from pda 1 is for ack
    of speed data.
    ismallsize = 10000
    ilargesize = 50000
```

```

ndatasets = 1 ' 390

createdata = 1 ' if createdata =1 then create test data

iwait = 0 ' iwait =0 data has been sent iwait=1 data is sending

istop = 0 ' istop =0 let loop run istop =1 stop the loop

nospeeddata = 1 ' nospeeddata=0 means dont compute speed info set it to 1000

ReDim dataheader(19)

filename = "Default.txt"

Dir1 = Drive1

If Len(Dir1) < 4 Then

    Text1 = Dir1 + filename

Else

    Text1 = Dir1 + "\" + filename

End If

Label1.Caption = "0"

Label2.Caption = GetWinsockState

Label3.Caption = "0"

SocketsInitialize

command1.Caption = "Listen"

Command2.Caption = "Send Small"

Command3.Caption = "Close Connections"

Command4.Caption = "Send Large"

Command5.Caption = "Send sm Cont"

Command6.Caption = "Send lg Cont"

Command7.Caption = "Stop Cont"

Command2.Enabled = False

```

```

Command3.Enabled = False

Command4.Enabled = False

Command5.Enabled = False

Command6.Enabled = False

Command7.Enabled = False

Timer1.Interval = 10 ' Set interval of rate measuring timer

Timer2.Interval = 1000 ' delay timer

Timer1.Enabled = False

Timer2.Enabled = False

timecount = 0

If createdata = 0 Then

' load small data packet

ismallsum = 0

Open "smdata.dat" For Binary Access Read As #1

For i = 1 To 20

    Get #1, i, dataheader(i - 1) ' get size and sum info

Next i

For i = 1 To 9

    sdata = sdata + Chr(dataheader(i - 1)) 'seprate size info

Next i

ismallsize = Val(sdata)

ReDim bsmallData(ismallsize - 1)

sdata = ""

For i = 10 To 19

```

```

        sdata = sdata + Chr(dataheader(i - 1)) ' seprate sum info
Next i

ismallsum = Val(sdata)

isum = 0

For i = 1 To ismallsize

    Get #1, i + 20, ibyte ' load small test data

    isum = isum + ibyte

    bsmallData(i - 1) = ibyte

Next i

Close #1

' get large data

Open "lgdata.dat" For Binary Access Read As #1

For i = 1 To 20

    Get #1, i, dataheader(i - 1) ' get size and sum info

Next i

sdata = ""

For i = 1 To 9

    sdata = sdata + Chr(dataheader(i - 1)) 'seprate size info

Next i

ilargesize = Val(sdata)

ReDim blargeData(ilargesize - 1)

sdata = ""

For i = 10 To 19

    sdata = sdata + Chr(dataheader(i - 1)) ' seprate sum info

Next i

ilargesum = Val(sdata)

```

```

isum = 0

For i = 1 To ilargesize

    Get #1, i + 20, ibyte ' load small test data

    isum = isum + ibyte

    blargeData(i - 1) = ibyte

Next i

Close #1

Else

ReDim bsmallData(ismallsize - 1)

ReDim blargeData(ilargesize - 1)

create_data

End If

End Sub

Private Sub Form_Unload(Cancel As Integer)

    SocketsCleanup

End Sub

Private Sub Command1_Click()

Rem Make socket connection

    Winsock1.Protocol = sckTCPProtocol

    Winsock1.LocalPort = 5149

    Winsock1.RemotePort = 5150

Rem    Winsock3.LocalPort = 6000

```

```

Rem    Winsock3.RemotePort = 6001

    command1.Enabled = False 'connect

    Winsock1.Listen

Rem    Winsock3.Listen

    timecount = 0

    Label2.Caption = GetWinsockState

End Sub

Private Sub Command2_Click()

Rem small data set send

    Dim bytessent As Long

    Dim bytesremaining As Long

    Dim test As String

    Dim i As Long

    Dim j As Integer

    disable_all_buttons ' call disable_all_buttons

    Open Text1 For Append As #1

    isum = (ismallsum + 20) * (ndatasets - 1) + ismallsum

    isize = ismallsize

    iwitchtest = 2

    await = 1 ' setup to wait until hand held has send final acknowledgement

    timecount = 0

    igettick = 0 ' set igettick =0 so only tick at first responce is recorded

    iget1tick = 0 ' set iget1tick =0 so only tick at first winsock2 "10" is recorded

```



```

iget2tick = 0 ' set iget1tick =0 so only tick at first winsock2 "20" is recorded
If ndatasets > 1 Then bsmallData(ismallsize - 1) = CByte(20)

tickstart = GetTickCount()

Rem    Timer1.Enabled = True

If ndatasets > 1 Then
    For idataset = 1 To ndatasets - 1
        Winsock2.SendData bsmallData()

        DoEvents

    Next idataset

    bsmallData(ismallsize - 1) = CByte(0)

End If

Winsock2.SendData bsmallData()

DoEvents

Do While iwait = 1
    Label2.Caption = Str(bytesremaining)

    DoEvents

Loop

Close #1

Label3.Caption = Str(isum)

End Sub

Private Sub Command3_Click()

```

```
Rem Close socket connections

    Winsock1.Close

Rem    Winsock2.Close

    Label2.Caption = GetWinsockState

    command1.Enabled = True 'connect

    Command2.Enabled = False ' send small

    Command3.Enabled = False ' disconnect

    Command4.Enabled = False ' send large

    Command5.Enabled = False ' send small cont

    Command6.Enabled = False ' send large cont

    Command7.Enabled = False ' stop send

End Sub
```

```
Private Sub Text1_Change()

    Dim ATEMP As String

    Dim l1 As Integer

    Dim l2 As Integer

    Rem If Len(Dir1) < 4 Then

    Rem ATEMP = Dir1

    Rem Else

    Rem ATEMP = Dir1

    Rem End If
```

```

Rem l1 = Len(ATEMP)

Rem l2 = Len(Text1)

Rem filename = Right(Text1, l2 - l1 - 1)

End Sub

Private Sub Timer1_Timer()
timecount = timecount + 1
End Sub

Private Sub Timer2_Timer()
iwait2 = 0
End Sub

Private Sub Winsock1_ConnectionRequest(ByVal requestID As Long)
    Winsock2.Accept requestID

    command1.Enabled = False 'connect
    Command2.Enabled = True ' send small
    Command3.Enabled = True ' disconnect
    Command4.Enabled = True ' send large
    Command5.Enabled = True ' send small cont
    Command6.Enabled = True ' send large cont
    Command7.Enabled = False ' stop send
    Label2.Caption = "Listening: Connected"
End Sub

```

```

Private Sub SocketsInitialize()

    Dim WSAD As WSADATA

    Dim iReturn As Integer

    Dim sLowByte As String, sHighByte As String, sMsg As String

    iReturn = WSASStartup(WS_VERSION_REQD, WSAD)

    If iReturn <> 0 Then

        MsgBox "Winsock.dll is not responding."

        End

    End If

    If lobyte(WSAD.wversion) < WS_VERSION_MAJOR Or _
        (lobyte(WSAD.wversion) = _
        WS_VERSION_MAJOR And hibernate(WSAD.wversion) < WS_VERSION_MINOR) Then

        sHighByte = Trim$(Str$(hibyte(WSAD.wversion)))

        sLowByte = Trim$(Str$(lobyte(WSAD.wversion)))

        sMsg = "Windows Sockets version " & sLowByte & "." & sHighByte

        sMsg = sMsg & " is not supported by winsock.dll "

        MsgBox sMsg

        End

    End If

    'iMaxSockets is not used in Winsock 2, so the following check is only
    'necessary for Winsock 1. If Winsock 2 is requested,

```

'the following check can be skipped.

```
If WSAD.iMaxSockets < MIN_SOCKETS_REQD Then
    sMsg = "This application requires a minimum of "
    sMsg = sMsg & Trim$(Str$(MIN_SOCKETS_REQD)) & " supported sockets."
    MsgBox sMsg
    End
End If

End Sub
```

```
Private Function GetWinsockState() As String
    Select Case Winsock1.State
        Case 0 'sckClosed
            GetWinsockState = "Closed"
        Case 1 'sckOpen
            GetWinsockState = "Open"
        Case 2 'sckListening
            GetWinsockState = "Listening"
        Case 3 'sckConnectionPending
            GetWinsockState = "Connection pending"
        Case 4 'sckResolvingHost
            GetWinsockState = "Resolving host"
        Case 5 'sckHostResolved
            GetWinsockState = "Host resolved"
        Case 6 'sckConnecting
```

```

        GetWinsockState = "Connecting"

    Case 7 'sckConnected

        GetWinsockState = "Connected"

    Case 8 'sckClosing

        GetWinsockState = "Peer is closing the connection"

    Case 9 'sckError

        GetWinsockState = "Error"

End Select

End Function

Private Function GetHostIP() As String

    Dim sHostName As String * 256

    Dim lHostEnt_Addr As Long

    Dim Host As HOSTENT

    Dim lHostIP_Addr As Long

    Dim bTempIP_Addr() As Byte

    Dim i As Integer

    Dim sIP_Addr As String

    If gethostname(sHostName, 256) = SOCKET_ERROR Then

        MsgBox "Windows Sockets error " & Str(WSAGetLastError())

        Exit Function

    Else

        sHostName = Trim$(sHostName)

    End If

```

```

lHostEnt_Addr = gethostbyname(sHostName)

If lHostEnt_Addr = 0 Then
    MsgBox "Winsock.dll is not responding."
    Exit Function
End If

RtlMoveMemory Host, lHostEnt_Addr, LenB(Host)
RtlMoveMemory lHostIP_Addr, Host.hAddrList, 4

'Get all of the IP addresses if the computer is multi-homed.
Do
    Redim bTempIP_Addr(1 To Host.hLength)
    RtlMoveMemory bTempIP_Addr(1), lHostIP_Addr, Host.hLength

    For i = 1 To Host.hLength
        sIP_Addr = sIP_Addr & bTempIP_Addr(i) & "."
    Next

    sIP_Addr = Mid$(sIP_Addr, 1, Len(sIP_Addr) - 1)
    GetHostIP = sIP_Addr
    sIP_Addr = ""
    Host.hAddrList = Host.hAddrList + LenB(Host.hAddrList)
    RtlMoveMemory lHostIP_Addr, Host.hAddrList, 4
Loop While (lHostIP_Addr <> 0)

```

```
End Function
```

```
Private Sub SocketsCleanup()
```

```
    Dim lReturn As Long
```

```
    lReturn = WSACleanup()
```

```
    If lReturn <> 0 Then
```

```
        MsgBox "Socket error " & Trim$(Str$(lReturn)) & _  
            " occurred in cleanup."
```

```
    End
```

```
End If
```

```
End Sub
```

```
Private Sub Winsock2_DataArrival(ByVal bytesTotal As Long)
```

```
    If igettick = 0 Then ticktemp = GetTickCount: igettick = 1
```

```
    Winsock2.GetData sdata
```

```
    ATEMP = ATEMP + sdata
```

```
    ATEMP2 = Right(sdata, 1)
```

```
    If ATEMP2 = "Z" Then ' when we get the Z we have gotten all the  
        data back form the client
```

```
        sdata = Left(ATEMP, Len(ATEMP) - 1) ' chop off end 0
```

```
        ATEMP = ""
```

```
    If Right(sdata, 1) = "1" Then ' get time to recieve data
```

```
        tickstop = ticktemp
```

```
        ctime = Val(Left(sdata, Len(sdata) - 1)) ' get time to recieve data
```

```
        Rem itrate2 = isize * ndatasets * 1000 / ctime
```



```

Winsock2.SendData Str(tickmiddle - tickstart) +
    " " + Str(ctime) + " " + Str(tickstop - tickstart
    - ctime) + Chr(0) ' send out the transmit speed
Print #1, Time, tickmiddle - tickstart, ctime, tickstop -
    tickstart - ctime, tickstop - tickstart
End If

If sdata = "2" Then Command2_Click ' send small
If sdata = "3" Then Command3_Click
If sdata = "4" Then Command4_Click ' send large
If sdata = "5" Then Command5_Click ' send small cont
If sdata = "6" Then Command6_Click ' send large cont
If sdata = "7" Then ' stop sending data
    Command7_Click
    Do Until istop = 1
        DoEvents
    Loop
End If

If Right(sdata, 1) = "8" Then ' change name command
    afilename = Left(sdata, Len(sdata) - 1)
    If Len(Dir1) < 4 Then
        Text1 = Dir1 + afilename
    Else
        Text1 = Dir1 + "\" + afilename
    End If
    sdata = ""
End If

```

```

    If sdata = "9" Then ' final acknowledgement
        tickstop = ticktemp
        Rem ctime = Val(Left(sdata, Len(sdata) - 1)) ' get time to receive data
        Print #1, Time, tickstop - tickstart
        await = 0 ' stop waiting for data back from handheld
    End If
End If

End Sub

Private Sub Winsock2_Error(ByVal Number As Integer, Description As String,
    ByVal Scode As Long, ByVal Source As String, ByVal HelpFile As String,
    ByVal HelpContext As Long, CancelDisplay As Boolean)
    DoEvents
End Sub

Private Sub Winsock2_SendComplete()
    tickmiddle = GetTickCount()
    Rem If idataset = ndatasets Then Timer1.Enabled = False: tickstop =
    GetTickCount(): Text1.Text = Str(tickstop - tickstart)
    If idataset = ndatasets Then tickmiddle = GetTickCount(): timecount
    = Str(tickmiddle - tickstart)
    If nospeeddata = 0 Then
        isendrate = 1000
    Else
        Rem If idataset = ndatasets Then isendrate = (isize * ndatasets /

```

```

timecount) * 100

        If idataset = ndatasets Then isendrate = (isize * ndatasets /
timecount) * 1000

    End If

    Label1.Caption = Str(isendrate)

    If iwichtest = 2 Then Command2.Enabled = True

    If iwichtest = 4 Then Command4.Enabled = True

    Command7.Enabled = True

    Command3.Enabled = True

    If idataset = ndatasets Then idataset = 0

End Sub

Sub create_data()

Dim i As Long

Dim j As Integer

    ismallsum = 0

    For i = 1 To ismallsize - 1

        j = Int(Rnd() * 245) + 10

        ismallsum = ismallsum + j

        bsmallData(i - 1) = CByte(j)

    Next i

    bsmallData(ismallsize - 1) = 0

    ilargesum = 0

    For i = 1 To ilargesize - 1

        j = Int(Rnd() * 245) + 10

```

```

        ilargesum = ilargesum + j
        blargeData(i - 1) = CByte(j)
    Next i
    blargeData(ilargesize - 1) = 0
End Sub

Private Sub Winsock2_SendProgress(ByVal bytesent As Long, ByVal bytesremaining As Long)
Label2.Caption = Str(bytesremaining)
DoEvents
End Sub

Sub disable_all_buttons()

    command1.Enabled = False 'connect
    Command2.Enabled = False ' send small
    Command3.Enabled = False ' disconnect
    Command4.Enabled = False ' send large
    Command5.Enabled = False ' send small cont
    Command6.Enabled = False ' send large cont
    Command7.Enabled = False ' stop send
    Text1.Enabled = False

End Sub

Sub enable_all_buttons()

```

```
command1.Enabled = False 'connect
Command2.Enabled = True ' send small
Command3.Enabled = True ' disconnect
Command4.Enabled = True ' send large
Command5.Enabled = True ' send small cont
Command6.Enabled = True ' send large cont
Command7.Enabled = False ' stop send
Text1.Enabled = True
```

```
End Sub
```

```
Private Sub Winsock3_ConnectionRequest(ByVal requestID As Long)
```

```
    If Winsock3.State <> sckClosed Then
        Winsock3.Close
    End If
    ' Accept the request with the requestID
    ' parameter.
    Winsock3.Accept requestID
```

```
End Sub
```

```
Private Sub Winsock3_DataArrival(ByVal bytesTotal As Long)
```

```
    Winsock3.GetData sdata2
    If iget1tick = 0 Then
        tickfirstrecv = GetTickCount()
        Label12.Caption = sdata2 + Str(tickfirstrecv)
    End If
```

```

    Rem If iget2tick = 0 And sdata2 = "20" Then

    Rem  tickrecvdone = GetTickCount()

    Rem  Label2.Caption = sdata2 + Str(tickrecvdone)

    Rem  End If

End Sub

```

A.3 Client Side Code

The code for the client side version of the transfer program is included below. The code is in Embedded Visual Basic and was edited using Embedded Visual Basic Studio 3.0. The file client.ebp is the Embedded Visual Basic Project file which basically provides header information. The file client.bas provides some basic functions that are used by the client.ebf file. The file client.ebf is the Embedded Visual Basic Form file which contains all the methods, control functions, and variables.

A.3.1 client.ebp.

```

Type=Exe

PlatformGUID={6D5C6210-E14B-11D2-B72A-0000F8026CEE}

DeviceGUID={3CFA6F81-EB79-11D2-BAC5-006097BA8DF0}

RemotePath=\Windows\Start Menu\Project1.vb

UpdateType=1

ForceRuntime=0

ForceComponent=0

Reference=*\G{00020430-0000-0000-C000-000000000046}#2.0#0#..\..\WINNT\System32\stdole2.tlb#OLE Au

Form=client.ebf

Object={F7346713-70C5-11D1-9AC9-00C04FAD5AEC}#1.0#0; msceimage.dll

```

Object={23CE4CF5-25A1-11D1-9A72-00A0C986B84A}#1.0#0; mscewinsock.dll

Module=Module1; client.bas

IconForm="Form1"

Startup="Form1"

ExeName32="client.vb"

Command32=""

Name="Project1"

HelpContextID="0"

CompatibleMode="0"

MajorVer=1

MinorVer=0

RevisionVer=0

AutoIncrementVer=0

ServerSupportFiles=0

VersionCompanyName="Microsoft Corporation"

CompilationType=0

OptimizationType=0

FavorPentiumPro(tm)=0

CodeViewDebugInfo=0

NoAliasing=0

BoundsCheck=0

OverflowCheck=0

FlPointCheck=0

FDIVCheck=0

UnroundedFP=0

StartMode=0

Unattended=0

Retained=0

ThreadPerObject=0

MaxNumberOfThreads=1

[OtherInfo]

ProjectType=WinCE

Platform={6D5C6210-E14B-11D2-B72A-0000F8026CEE}

A.3.2 *client.bas.*

Attribute VB_Name = "Module1"

Option Explicit

```
Public Declare Function CreateFile Lib "Coredll" Alias "CreateFileW" ( _  
    ByVal lpFileName As String, _  
    ByVal dwDesiredAccess As Long, _  
    ByVal dwShareMode As Long, _  
    lpSecurityAttributes As Long, _  
    ByVal dwCreationDisposition As Long, _  
    ByVal dwFlagsAndAttributes As Long, _  
    ByVal hTemplateFile As Long) As Long
```

```
Public Declare Function ReadFile Lib "Coredll" ( _  
    ByVal hFile As Long, _  
    ByVal lpBuffer As String, _
```



```

    ByVal nNumberOfBytesToRead As Long, _
    lpNumberOfBytesRead As Long, _
    ByVal lpOverlapped As Long) As Long

Public Declare Function WriteFile Lib "Coredll" ( _
    ByVal hFile As Long, _
    ByVal lpBuffer As String, _
    ByVal nNumberOfBytesToWrite As Long, _
    lpNumberOfBytesWritten As Long, _
    ByVal lpOverlapped As Long) As Long

Public Declare Function CloseHandle Lib "Coredll" ( _
    ByVal hObject As Long) As Long

Public Declare Function GetLastError Lib "Coredll" () As Long

Public Declare Function LoadCursor Lib "Coredll" _
    Alias "LoadCursorW" ( _
    ByVal hInstance As Long, _
    ByVal lpCursorName As Long) As Long

Public Declare Function SetCursor Lib "Coredll" ( _
    ByVal hCursor As Long) As Long

Public Const READ_CONTROL = &H20000

Public Const READ_WRITE = 2

```

```

Public Const FILE_READ_DATA = (&H1)

Public Const FILE_READ_ATTRIBUTES = (&H80)

Public Const FILE_READ_EA = (&H8)

Public Const FILE_WRITE_ATTRIBUTES = (&H100)

Public Const FILE_WRITE_DATA = (&H2)

Public Const FILE_WRITE_EA = (&H10)

Public Const FILE_APPEND_DATA = (&H4)

Public Const SYNCHRONIZE = &H100000

Public Const CREATE_ALWAYS = 2

Public Const OPEN_EXISTING = 3

Public Const OPEN_ALWAYS = 4

Public Const STANDARD_RIGHTS_WRITE = &H20000

Public Const STANDARD_RIGHTS_READ = &H20000

Public Const GENERIC_READ = &H80000000

Public Const GENERIC_WRITE = &H40000000

Public Const IDC_WAIT = 32514

Public Function WaitCursor(bWait As Boolean) As Long

    Dim hCursor As Long

    'Obtain the handle to the cursor.

    If bWait Then

        'Get handle to the wait cursor.

        hCursor = LoadCursor(0, IDC_WAIT)

```

```
Else
    'Restore default cursor.
    hCursor = LoadCursor(0, 0)
End If

'Set the cursor based on the cursor handle.
WaitCursor = SetCursor(hCursor)
```

```
End Function
```

A.3.3 client.ebf.

```
VERSION 5.00
```

```
Object = "{23CE4CF5-25A1-11D1-9A72-00A0C986B84A}#1.0#0"; "mscwinsock.dll"
```

```
Begin VB.Form Form1
```

```
    Appearance      = 0 'Flat
    BackColor        = &H80000005&
    Caption          = "client"
    ClientHeight     = 4455
    ClientLeft       = 60
    ClientTop        = 840
    ClientWidth      = 5895
    ForeColor        = &H80000008&
    ScaleHeight      = 4455
    ScaleWidth       = 5895
    ShowOK           = -1 'True
```

```
Begin WinSockCtl.WinSock WinSock2
```

```

    Left          = 2520
    Top           = 3960
    _cx           = 1800
    _cy           = 1000
    LocalPort     = 0
    Protocol      = 0
    RemoteHost    = ""
    RemotePort    = 0
    ServiceName   = ""
End

Begin WinSockCtl.WinSock WinSock1

    Left          = 120
    Top           = 3840
    _cx           = 1800
    _cy           = 1000
    LocalPort     = 0
    Protocol      = 0
    RemoteHost    = ""
    RemotePort    = 0
    ServiceName   = ""
End

Begin VBCE.Timer Timer2

    Left          = 3600
    Top           = 3240
    _cx           = 847
    _cy           = 847

```

```

    Enabled      = -1 'True
    Interval     = 0
End
Begin VBCE.TextBox Text2
    Height       = 375
    Left         = 240
    TabIndex     = 10
    Top          = 360
    Width        = 2895
    _cx         = 5106
    _cy         = 661
    BackColor    = -2147483643
    BorderStyle  = 1
    Enabled      = -1 'True
BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
    Name         = "Tahoma"
    Size         = 8.25
    Charset      = 0
    Weight       = 400
    Underline    = 0 'False
    Italic       = 0 'False
    Strikethrough = 0 'False
EndProperty
ForeColor      = -2147483640
Text           = "Text2"
Alignment      = 0

```

```

HideSelection = -1 'True
Locked        = 0  'False
MaxLength     = 0
MultiLine    = 0  'False
PasswordChar  = ""
ScrollBars    = 0

End

Begin VBCE.Timer Timer1
    Left       = 1800
    Top        = 3840
    _cx        = 847
    _cy        = 847
    Enabled    = -1 'True
    Interval   = 0
End

Begin VBCE.CommandButton Command8
    Height     = 495
    Left       = 1800
    TabIndex   = 9
    Top        = 840
    Width      = 1335
    _cx        = 2355
    _cy        = 873
    BackColor  = 12632256
    Caption    = "change name"
    Enabled    = -1 'True

```

```

BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
    Name          =    "Tahoma"
    Size          =    8.25
    Charset       =    0
    Weight        =    400
    Underline     =    0    'False
    Italic        =    0    'False
    Strikethrough =    0    'False
EndProperty
Style            =    0
End
Begin VBCE.TextBox Text1
    Height        =    495
    Left          =    240
    TabIndex      =    8
    Top           =    3240
    Width         =    2895
    _cx          =    5106
    _cy          =    873
    BackColor     =    -2147483643
    BorderStyle   =    1
    Enabled       =    -1    'True
BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
    Name          =    "Tahoma"
    Size          =    8.25
    Charset       =    0

```

```

        Weight          = 400
        Underline       = 0   'False
        Italic          = 0   'False
        Strikethrough   = 0   'False
    EndProperty
    ForeColor          = -2147483640
    Text               = "Text1"
    Alignment          = 0
    HideSelection      = -1   'True
    Locked             = 0   'False
    MaxLength          = 0
    MultiLine          = 0   'False
    PasswordChar       = ""
    ScrollBars         = 0
End
Begin VBCE.CommandButton Command7
    Height             = 495
    Left               = 1800
    TabIndex           = 7
    Top                = 2640
    Width              = 1335
    _cx                = 2355
    _cy                = 873
    BackColor          = 12632256
    Caption            = "stop"
    Enabled            = -1   'True

```



```

BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
    Name          =    "Tahoma"
    Size          =    8.25
    Charset       =    0
    Weight        =    400
    Underline     =    0    'False
    Italic        =    0    'False
    Strikethrough =    0    'False
EndProperty
Style            =    0
End
Begin VBCE.CommandButton Command6
    Height        =    495
    Left          =    1800
    TabIndex      =    6
    Top           =    2040
    Width         =    1335
    _cx          =    2355
    _cy          =    873
    BackColor     =    12632256
    Caption       =    "Send lg cont"
    Enabled       =    -1    'True
BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
    Name          =    "Tahoma"
    Size          =    8.25
    Charset       =    0

```

```

        Weight          = 400
        Underline       = 0   'False
        Italic          = 0   'False
        Strikethrough   = 0   'False
    EndProperty
    Style              = 0
End
Begin VBCE.CommandButton Command5
    Height            = 495
    Left              = 1800
    TabIndex          = 5
    Top               = 1440
    Width             = 1335
    _cx               = 2355
    _cy               = 873
    BackColor         = 12632256
    Caption           = "Send sm cont"
    Enabled           = -1   'True
BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
    Name              = "Tahoma"
    Size              = 8.25
    Charset           = 0
    Weight            = 400
    Underline         = 0   'False
    Italic            = 0   'False
    Strikethrough     = 0   'False

```

```

EndProperty

Style          = 0

End

Begin VBCE.CommandButton Command4

Height        = 495

Left          = 240

TabIndex      = 4

Top           = 2040

Width         = 1335

_cx           = 2355

_cy           = 873

BackColor     = 12632256

Caption       = "Send mid cont"

Enabled       = -1 'True

BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}

Name          = "Tahoma"

Size          = 8.25

Charset       = 0

Weight        = 400

Underline     = 0 'False

Italic        = 0 'False

Strikethrough = 0 'False

EndProperty

Style          = 0

End

Begin VBCE.CommandButton Command2

```

```

Height          = 495
Left            = 240
TabIndex        = 3
Top             = 1440
Width           = 1335
_cx             = 2355
_cy             = 873
BackColor       = 12632256
Caption         = "Send Small"
Enabled         = -1 'True
BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
    Name         = "Tahoma"
    Size         = 8.25
    Charset      = 0
    Weight       = 400
    Underline    = 0 'False
    Italic       = 0 'False
    Strikethrough = 0 'False
EndProperty
Style           = 0
End
Begin VBCE.Label Label1
    Height       = 375
    Left         = 240
    TabIndex     = 2
    Top          = 0

```

```

Width          = 1935
_cx            = 3413
_cy            = 661
AutoSize       = 0   'False
BackColor      = -2147483643
BackStyle      = 1
BorderStyle    = 0
Caption        = "Label1"
Enabled        = -1  'True
BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
    Name        = "Tahoma"
    Size        = 8.25
    Charset     = 0
    Weight      = 400
    Underline   = 0   'False
    Italic      = 0   'False
    Strikethrough = 0   'False
EndProperty
ForeColor      = -2147483640
Alignment      = 0
UseMnemonic    = -1  'True
WordWrap       = 0   'False
End
Begin VBCE.CommandButton Command3
    Height      = 495
    Left        = 240

```

```

TabIndex      = 1
Top           = 2640
Width        = 1335
_cx          = 2355
_cy          = 873
BackColor    = 12632256
Caption      = "disconnect"
Enabled      = -1 'True
BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
    Name      = "Tahoma"
    Size      = 8.25
    Charset   = 0
    Weight    = 400
    Underline = 0 'False
    Italic    = 0 'False
    Strikethrough = 0 'False
EndProperty
Style        = 0
End
Begin VBCE.CommandButton Command1
    Height    = 495
    Left      = 240
    TabIndex  = 0
    Top       = 840
    Width     = 1335
    _cx      = 2355

```

```

        _cy            = 873
        BackColor      = 12632256
        Caption        = "connect"
        Enabled        = -1 'True
        BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
            Name        = "Tahoma"
            Size        = 8.25
            Charset     = 0
            Weight      = 400
            Underline   = 0 'False
            Italic      = 0 'False
            Strikethrough = 0 'False
        EndProperty
        Style          = 0
    End

End

Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Dim ireccount As Integer
Dim irecvcount As Integer
Dim inBytes() As Byte

```

```
Dim MyString As String

Dim bLast As Boolean

Dim bNotFirst As Boolean

Dim hFile As Long

Dim iwitchtest As Integer

Dim iwait As Integer

Dim aremotehost As String

Dim itestdata As Integer

Dim atempstring As String

Dim i As Long

Dim isum As Long

Dim winsockerror As Integer

Dim winsocktest As Integer

Dim msg As String

Dim ibytecount As Long

Dim dataset As Integer

Declare Function GetTickCount Lib "Coredll" () As Long

Declare Function Sleep Lib "Coredll" (ByVal Sleep As Long) As Long
```



```

Private Sub Command1_Click()

winsocktest = 1

    WaitCursor True

    Label1.Caption = "port setup"

    WinSock1.LocalPort = 5150

    WinSock1.RemotePort = 5149

Rem    WinSock2.LocalPort = 6001

Rem    WinSock2.RemotePort = 6000

    aremotehost = Text2

    Label1.Caption = "ip setup"

    WinSock1.RemoteHost = aremotehost 'Put your IP address here

Rem    WinSock2.RemoteHost = aremotehost

    Label1.Caption = "BEFORE connect"

    WinSock1.Connect

Rem    WinSock2.Connect

    Label1.Caption = "Connected"

    enable_all_buttons

    WaitCursor False

End Sub

Private Sub Command3_Click()

Rem disconnect

    WinSock1.SendData "3Z"

```

```
        disable_all_buttons
        Timer1.Enabled = True

End Sub

Private Sub Command2_Click()
Rem small
        isum = 0
        ibytecount = 0
        ireccount = 0
        dataset = 0
        iwitchtest = 2
        disable_all_buttons
        WinSock1.SendData "2Z"

End Sub

Private Sub Command4_Click()
Rem large
        isum = 0
        ibytecount = 0
        ireccount = 0
        dataset = 0
        iwitchtest = 4
        disable_all_buttons
        WinSock1.SendData "4Z"
```

```
End Sub
```

```
Private Sub Command5_Click()
```

```
Rem sm cont
```

```
    isum = 0
```

```
    ibytecount = 0
```

```
    ireccount = 0
```

```
    dataset = 0
```

```
    iwitchtest = 5
```

```
    disable_all_buttons
```

```
    WinSock1.SendData "5Z"
```

```
End Sub
```

```
Private Sub Command6_Click()
```

```
Rem lg cont
```

```
    isum = 0
```

```
    ibytecount = 0
```

```
    ireccount = 0
```

```
    dataset = 0
```

```
    iwitchtest = 6
```

```
    disable_all_buttons
```

```
    WinSock1.SendData "6Z"
```

```
End Sub
```

```
Private Sub Command7_Click()
```

```
Rem stop
```

```

iwitchtest = 7

If iwait = 0 Then

    WinSock1.SendData "7Z" ' if not waiting for data send stop button

    enable_all_buttons

End If

End Sub

Private Sub Command8_Click()

Rem change data file name

    Dim atemp As String

    atemp = Text1 + "8Z"

    WinSock1.SendData atemp

End Sub

Private Sub Form_Load()

msg = String(18, Chr(0))

winsocketest = 0

    Timer1.Interval = 1000

    Command3.Caption = "Close Winsock"

    disable_all_buttons

    Command1.Enabled = True

    Label1.Caption = ""

    ireccount = 0

    iwait = 0

```

```
    iwitchtest = 0

    isum = 0

    Rem aremotehost = "134.131.109.135"

    aremotehost = "192.131.110.160"

    Text2 = aremotehost

End Sub
```

```
Private Sub Label1_Click()
```

```
End Sub
```

```
Private Sub Timer1_Timer()
```

```
    WinSock1.Close

    Rem    winsock2.close

    Command1.Enabled = True

    Timer1.Enabled = False

End Sub
```

```
Private Sub Timer2_Timer()
```

```
    iwait = 0

End Sub
```

```
Private Sub WinSock1_Connect()
```

```
End Sub
```

```

Private Sub WinSock1_ConnectionRequest()

End Sub

Private Sub WinSock1_DataArrival(ByVal bytesTotal As Long)

    Command7.Enabled = False

    ReDim inBytes(bytesTotal)

    Dim tickstart As Long

    Dim tickstop As Long

    iwait = 1

    irecvcount = irecvcount + 1

    Do While WinSock1.BytesReceived > 0

        WinSock1.GetData inBytes, (vbByte + vbArray)

        ireccount = ireccount + 1

        Rem Label1.Caption = CStr(ireccount)

    Rem For i = 1 To UBound(inBytes) + 1

    Rem ibytecount = ibytecount + 1

    Rem isum = isum + inBytes(i - 1)

    Rem Next i

    Rem iwait = iwait

    Loop

    If inBytes(UBound(inBytes)) = 0 Then

        Rem Text2.Text = CStr(isum)

```

```

Command7.Enabled = True

atempstring = ""

dataset = dataset + 1

iwait = 0

If iwithctest = 2 Then Command2.Enabled = True

If iwithctest = 4 Then Command4.Enabled = True

If iwithctest = 7 Then

    enable_all_buttons

    WinSock1.SendData "7Z"

    dataset = 0

Else

    WinSock1.SendData "9Z" ' send final acknowledgement

End If

If iwithctest = 0 Then enable_all_buttons

Text2.Text = CStr(dataset)

ireccount = 0

irecvcount = 0

End If

End Sub

Private Sub Form_OKClick()

    App.End

End Sub

Sub disable_all_buttons()

    Command1.Enabled = False

```

```
Command2.Enabled = False
Command3.Enabled = False
Command4.Enabled = False
Command5.Enabled = False
Command6.Enabled = False
Command7.Enabled = False
Command8.Enabled = False
End Sub
```

```
Sub enable_all_buttons()

Command1.Enabled = False
Command2.Enabled = True
Command3.Enabled = True
Command4.Enabled = True
Command5.Enabled = True
Command6.Enabled = True
Command7.Enabled = False
Command8.Enabled = True

End Sub
```

```
Private Sub WinSock1_Error(ByVal number As Long, ByVal description As String)
winsockerror = 111
If winsocktest = 1 Then
Stop
End If
```



```
Label1.Caption = number
```

```
End Sub
```

```
Private Sub WinSock1_SendComplete()
```

```
    iwait = 0
```

```
End Sub
```

```
Private Sub CEDoEvents()
```

```
    Dim PM_Remove As Integer
```

```
    Label1.Caption = "IN do events"
```

```
    PM_Remove = 1
```

```
    If peekmessage(msg, 0, 0, 0, PM_Remove) Then
```

```
        translatemessage (msg)
```

```
        dispatchmessage (msg)
```

```
    End If
```

```
End Sub
```

```
Private Sub WinSock1_SendProgress(ByVal bytesSent As Long, ByVal bytesRemaining As Long)
```

```
End Sub
```

```
Private Sub WinSock2_DataArrival(ByVal bytesTotal As Long)
```

```
End Sub
```

Appendix B. VPN-1 Configuration

The VPN-1/Firewall-1 [Ltd00] server software tested was version 4.1. The following steps were used to install and setup the software:

- 1) The autorun program from Checkpoint's VPN-1/Firewall-1 installation disk was used to install the server software. All installation options were left at the default values.

- 2) Then the network objects were defined.

- a) The server was added as a gateway, with FWZ, Checkpoint's native encryption algorithm, selected. The gateway interface was set as external, to allow communication with the external, mobile client.

- b) The PDA client was added, again with FWZ encryption enabled.

- 3) The policy was installed. Using the policy manager, a policy consisting of a single rule was created. This rule required all traffic in the network to be encrypted.

- 4) The client program was installed via the USB port to a laptop computer. Installation options were left at default values. Once installed, the gateway object was defined, establishing a relationship.

- 5) Once the server and client software was properly installed, the transfer program was ran to get the test data.

Appendix C. Raw Data

C.1 Introduction

The raw data recorded from the throughput and battery life experiments are included below.

C.2 Throughput

The data from the throughput experiment is in the following format:

Time file was sent	Duration of transfer
--------------------	----------------------

C.2.1 Test 1.

11:29:25 AM	6639
11:29:33 AM	6529
11:29:40 AM	6659
11:29:48 AM	6740
11:29:56 AM	6590
11:30:04 AM	6599
11:30:11 AM	6509
11:30:19 AM	6619
11:30:27 AM	6589
11:30:35 AM	6589
11:30:42 AM	6469
11:30:50 AM	6579
11:30:58 AM	6590
11:31:05 AM	6660
11:31:13 AM	6510

C.2.2 Test 2.

11:03:30 AM	34159
11:04:05 AM	33779
11:04:40 AM	33818
11:05:16 AM	34049
11:05:51 AM	33928
11:06:26 AM	34460
11:07:01 AM	33979
11:07:37 AM	34029
11:08:12 AM	34230
11:08:47 AM	33929
11:09:22 AM	34029
11:09:58 AM	34069
11:10:33 AM	33859
11:11:08 AM	34079
11:11:43 AM	34249

C.2.3 Test 3.

3:26:45 PM	69110
3:27:55 PM	68539
3:29:05 PM	68718
3:30:14 PM	68028
3:31:24 PM	68839

3:32:33 PM	68429
3:33:43 PM	68238
3:34:53 PM	68829
3:36:02 PM	68078
3:37:11 PM	68088
3:38:21 PM	69039
3:39:31 PM	68949
3:40:41 PM	68248
3:41:50 PM	68248
3:43:00 PM	68539

C.2.4 Test 4.

11:14:28 AM	6680
11:14:36 AM	6659
11:14:44 AM	6660
11:14:52 AM	6700
11:15:00 AM	6540
11:15:07 AM	6690
11:15:15 AM	6659
11:15:23 AM	6680
11:15:31 AM	6510
11:15:39 AM	6680
11:15:46 AM	6679
11:15:54 AM	6529
11:16:02 AM	6619

11:16:10 AM 6710

11:16:18 AM 6780

C.2.5 Test 5.

11:18:22 AM 34059

11:18:57 AM 34059

11:19:32 AM 34029

11:20:07 AM 34049

11:20:42 AM 33809

11:21:17 AM 33819

11:21:52 AM 33838

11:22:27 AM 33899

11:23:02 AM 34099

11:23:38 AM 33999

11:24:13 AM 33869

11:24:48 AM 33969

11:25:23 AM 34089

11:25:58 AM 33889

11:26:34 AM 34420

11:27:09 AM 33788

C.2.6 Test 6.

4:11:09 PM 68889

4:12:19 PM 69320

4:13:29 PM	68719
4:14:40 PM	69290
4:15:50 PM	68819
4:17:00 PM	69179
4:18:10 PM	68669
4:19:20 PM	69030
4:20:30 PM	68539
4:21:40 PM	69180
4:22:49 PM	68238
4:24:00 PM	69650
4:25:10 PM	68749
4:26:20 PM	68889
4:27:30 PM	68940

C.2.7 Test 7.

11:00:36 AM	6570
11:00:44 AM	6830
11:00:52 AM	6780
11:00:59 AM	6680
11:01:07 AM	6610
11:01:15 AM	6619
11:01:23 AM	6660
11:01:31 AM	6610
11:01:39 AM	6660
11:01:46 AM	6629

11:01:54 AM	6710
11:02:02 AM	6700
11:02:10 AM	6770
11:02:18 AM	6669
11:02:26 AM	6710

C.2.8 Test 8.

11:32:14 AM	33918
11:32:49 AM	34069
11:33:25 AM	34139
11:34:00 AM	33939
11:34:35 AM	34209
11:35:10 AM	33738
11:35:45 AM	33999
11:36:20 AM	33999
11:36:55 AM	33969
11:37:31 AM	34420
11:38:06 AM	34019
11:38:41 AM	33979
11:39:16 AM	34099
11:39:51 AM	33759
11:40:27 AM	34309

C.2.9 Test 9.

3:47:59 PM	68238
3:49:09 PM	69140
3:50:19 PM	68358
3:51:29 PM	68950
3:52:39 PM	68859
3:53:49 PM	68739
3:54:59 PM	68970
3:56:09 PM	69330
3:57:19 PM	68448
3:58:29 PM	68859
3:59:39 PM	68459
4:00:49 PM	69100
4:01:58 PM	68027
4:03:08 PM	68909
4:04:18 PM	68649

C.2.10 Test 10.

11:31:36 AM	6799
11:31:44 AM	6840
11:31:52 AM	6819
11:32:00 AM	6840
11:32:08 AM	6799
11:32:16 AM	6800
11:32:24 AM	6789
11:32:32 AM	6830

11:32:40 AM	6829
11:32:48 AM	6790
11:32:56 AM	6820
11:33:04 AM	6810
11:33:12 AM	6830
11:33:20 AM	6770
11:33:28 AM	6770
11:33:36 AM	6810
11:33:44 AM	6820
11:33:52 AM	6820

C.2.11 Test 11.

11:38:09 AM	34930
11:38:45 AM	34790
11:39:21 AM	34911
11:39:58 AM	35031
11:40:34 AM	35111
11:41:10 AM	35111
11:41:46 AM	35111
11:42:23 AM	35140
11:42:59 AM	34830
11:43:35 AM	35000
11:44:11 AM	35071
11:44:47 AM	34820
11:45:23 AM	34860

11:45:59 AM 34920

11:46:35 AM 34960

11:47:12 AM 35321

C.2.12 Test 12.

11:49:23 AM 71583

11:50:36 AM 71483

11:51:49 AM 71693

11:53:01 AM 70962

11:54:13 AM 71323

11:55:25 AM 71012

11:56:38 AM 71433

11:57:50 AM 71012

11:59:03 AM 71263

12:00:15 PM 71042

12:01:27 PM 71223

12:02:40 PM 71123

12:03:52 PM 71383

12:05:04 PM 71062

12:06:17 PM 71372

C.2.13 Test 13.

12:12:19 PM 6820

12:12:27 PM 6940

12:12:35 PM	6849
12:12:43 PM	6880
12:12:51 PM	6849
12:12:59 PM	6900
12:13:07 PM	6880
12:13:16 PM	6900
12:13:24 PM	6890
12:13:32 PM	6840
12:13:40 PM	6950
12:13:48 PM	6819
12:13:56 PM	6860
12:14:04 PM	6809
12:14:12 PM	6870

C.2.14 Test 14.

12:15:11 PM	35401
12:15:47 PM	35220
12:16:24 PM	35311
12:17:00 PM	35240
12:17:37 PM	35501
12:18:13 PM	35130
12:18:49 PM	35130
12:19:26 PM	35331
12:20:02 PM	35281
12:20:39 PM	35060

12:21:15 PM	35151
12:21:51 PM	35271
12:22:28 PM	35200
12:23:04 PM	35181
12:23:41 PM	35321

C.2.15 Test 15.

12:26:14 PM	71573
12:27:27 PM	71282
12:28:39 PM	71082
12:29:52 PM	71593
12:31:04 PM	71243
12:32:17 PM	71704
12:33:30 PM	71433
12:34:42 PM	71643
12:35:55 PM	71323
12:37:08 PM	71552
12:38:20 PM	71443
12:39:33 PM	71623
12:40:46 PM	71373
12:41:58 PM	71694
12:43:11 PM	71593

C.2.16 Test 16.

12:46:21 PM	6800
12:46:29 PM	6880
12:46:37 PM	6850
12:46:45 PM	6820
12:46:53 PM	6840
12:47:01 PM	6810
12:47:09 PM	6860
12:47:17 PM	6910
12:47:25 PM	6860
12:47:33 PM	6829
12:47:41 PM	6870
12:47:49 PM	6809
12:47:57 PM	6900
12:48:05 PM	6870
12:48:13 PM	6920

C.2.17 Test 17.

12:49:24 PM	35391
12:50:00 PM	35642
12:50:37 PM	35531
12:51:14 PM	35581
12:51:50 PM	35291
12:52:27 PM	35622
12:53:03 PM	35331
12:53:40 PM	35531

12:54:17 PM	35321
12:54:53 PM	35321
12:55:30 PM	35481
12:56:07 PM	35671
12:56:43 PM	35150
12:57:20 PM	35591
12:57:56 PM	35531

C.2.18 Test 18.

2:58:54 PM	73155
3:00:08 PM	72814
3:01:22 PM	72965
3:02:36 PM	72904
3:03:50 PM	72875
3:05:04 PM	72884
3:06:18 PM	72654
3:07:32 PM	72845
3:08:46 PM	73216
3:10:00 PM	72745
3:11:14 PM	72965
3:12:28 PM	72915
3:13:43 PM	73065
3:14:57 PM	72914
3:16:11 PM	72825

C.3 Battery Life

Table C-1 contains the data from the battery life experiment.

Table C.1 Battery Life est Data

Test		Combination			
		1	2	3	4
1	Start	11:52:28 AM	12:42:26 PM	9:08:40 AM	3:39:21 PM
	Stop	2:48:21 PM	3:10:33 PM	12:08:35 PM	6:34:29 PM
2	Start	2:07:27 PM	8:42:56 AM	3:48:43 PM	10:36:15 AM
	Stop	4:39:55 PM	11:04:57 AM	6:44:59 PM	1:26:45 PM
3	Start	11:34:13 AM	3:57:23 PM	12:17:08 PM	8:31:25 AM
	Stop	2:04:04 PM	6:24:38 PM	3:20:52 PM	11:16:19 AM
4	Start	2:46:23 PM	12:56:26 PM	8:53:02 AM	9:10:08 AM
	Stop	5:35:20 PM	3:20:24 PM	11:48:46 AM	11:55:30 AM
5	Start	12:40:08 PM	8:46:25 AM	4:24:09 PM	3:42:19 PM
	Stop	3:31:29 PM	11:10:38 AM	7:27:39 PM	6:31:55 PM

Appendix D. Laptop Performance

The results for test cases 1-9 ran on a Gateway 2000 Solo laptop computer are given in this Appendix. The same tables presented in Chapter 4 are provided. The PDA client transfer program was used, with slight modifications for the Windows 2000 environment (see Appendix A).

Table D.1 Laptop Results

Test	MEAN bps	STANDARD DEV. bps	C.O.V.	90% CONFIDENCE INTERVAL bps
1	1377975	154343	0.112	(1312425, 1443524)
2	1634635	105482	0.065	(1589837, 1679433)
3	1609249	94506	0.059	(1569112, 1649385)
4	1999997	13	0.000	(1999991, 2000002)
5	2211151	29813	0.013	(2198489, 2223813)
6	2193080	45943	0.021	(2173568, 2212592)
7	1947317	99068	0.051	(1905243, 1989391)
8	2113963	89114	0.042	(2076116, 2151809)
9	2117460	56927	0.027	(2093284, 2141636)

Table D.2 Laptop Computation of Effects

all values bps	CLOSE	MIDDLE	FAR	ROW MEAN	ROW EFFECT
SMALL	1377975	1999997	1947317	1775096	-136551
MEDIUM	1634635	2211151	2113963	1986583	74935
LARGE	1609249	2211151	2113963	1986583	61616
COLUMN MEAN	1540619	2134742	2059580	1911647	
COLUMN EFFECT	-371028	223095	147933		

Table D.3 Laptop Interactions

all values bps	CLOSE	MIDDLE	FAR
SMALL	-26094	1805	24288
MEDIUM	19080	1473	-20553
LARGE	7014	-3279	-3735

Table D.4 Laptop ANOVA

COMPONENT	SoS	PoV	DoF	MS	F-C	F-T
ALL MEANS	505007561415366		135			
MEAN RESPONSE	493343362377749		1			
ALL MEANS - MEAN RESPONSE	11664199037617	100	134			
CLIENT DISTANCE	9419278645526	81	2	4709639322763	624	≈ 2.4
WORKLOAD	1262612074586	11	2	631306037293	84	≈ 2.4
INTERACTIONS	32048631725	0	4	8012157931	13.91	≈ 2.1
ERRORS	950259685781	8	126	8012157931	1	
Standard Deviation for Errors = 10.26						

Table D.5 Laptop Confidence Intervals for Effects

all values bps	MEAN EFFECT	STANDARD DEVIATION	CONFIDENCE INTERVAL
MEAN	1911647	7474.27	(1899352, 1923942)
CLOSE	-371028	10570.22	(-3884160, -35363)
MIDDLE	223095	10570.22	(205737, 240483)
FAR	1479334	10570.22	(130545, 165321)
SMALL	-136551	10570.22	(-1539393, -119163)
MEDIUM	74935	10570.22	(57547, 92323)
LARGE	61616	10570.22	(44228, 79004)

Table D.6 Laptop Client Distance Impact

	CLOSE	MIDDLE	FAR
EFFECT	-371028 ± 12295 bps	2230950 ± 12295 bps	147933 ± 12295 bps
IMPACT	-19.41%	11.67%	7.74%
Mean Throughput			= 1911647 bps

Table D.7 Laptop Workload Impact

	SMALL	MEDIUM	LARGE
EFFECT	-136551 ± 12295 bps	74935 ± 12295 bps	61616 ± 12295 bps
IMPACT	-7.14%	3.92%	3.22%
Mean Throughput			= 1911647 bps

Bibliography

- [BB01] Jason Brooks and Herb Bethoney. The LAN, PAN, WAN plan. *eWeek*, 18(2):48–49, January 2001.
- [Bia00] Giuseppe Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, March 2000.
- [CG96] Harshal S. Chhaya and Sanjay Gupta. Throughput and fairness properties of asynchronous data transfer methods in the IEEE 802.11 MAC protocol. *Proceedings from the 6th International Symposium on Personal, Indoor, and Mobile Radio Communications*, 2:613–617, 1996.
- [Cis00] Cisco Systems, INC. Using the cisco aironet 340 series wireless bridge, 2000.
- [Com00] Compaq. iPAQ H3000 pocket PC reference guide, June 2000.
- [Cus01] Jeff Custard. Wireless VPN performance tests. <http://www.scd.ucar.edu/nets/projects/wireless/performance.test.vpn.htm>, January 2001.
- [Fre02] Dr. Craig C. Freudenrich. How personal digital assistants (PDAs) work. HowStuff-Works (<http://www.howstuffworks.com>), 2002.
- [Gar02] Dale Gardner. Wireless insecurities, control mobile computing vulnerabilities before they get control of you. *Information Security*, January 2002.
- [Jai91] Raj Jain. *The Art of Computer Systems Performance Analysis*. John Wiley & Sons, INC., 1991.
- [KK00] P. Krishnamurthy and J. Kabara. Security architecture for wireless residential networks. *52nd Vehicular Technology Conference*, 4:1960–1966, September 2000.
- [Kor98] L. Korba. Security systems for wireless local area networks. *The 9th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 3:1550–1554, September 1998.
- [Lab00] RSA Laboratories. RSA laboratories’ frequently asked questions about today’s cryptography, 2000. Version 4.1.
- [Ltd00] Checkpoint Software Technologies Ltd. Check point virtual private networks, August 2000.
- [MBWV98] U. Murthy, O. Bukhres, W. Winn, and E. Vanderdez. Firewalls for security in wireless networks. *Proceedings of the 31st Hawaii International Conference on System Sciences*, 7:672–680, January 1998.
- [MM99] A. Marincic and D. Milovanovic. Wireless local area networks. *4th International conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services*, 1:291–299, October 1999.
- [MW02] Alice M. Mulvehill and Randall Whitaker. Human Interaction with Software Agents (H.I.S.A.). Technical Report AFRL-HE-WP-TR-2002-0007, Human Effectiveness Directorate, Sustainment Logistics Branch, WPAFB, OH., 2002.
- [NT94] B.C. Neuman and T. Ts’o. Kerberos: an authentication service for computer networks. *IEEE Communications Magazine*, 32:33–38, September 1994. Issue 9.

- [Off01] Headquarters Standard Systems Group, Air Force Systems Networking Program Office. Common user virtual private network program overview. Internal Report, July 2001.
- [PE00] C.J.C. Pena and J. Evans. Performance evaluations of software virtual private networks. *25th Annual IEEE Annual IEEE Conference on Local Computer Networks*, pages 522–523, November 2000.
- [Rus01] S.F. Russell. Wireless network security for users. *International Conference on Information Technology: Coding and Computing*, pages 172–177, April 2001.
- [SBB01] Thomas M. Stricker, Armin Brunner, and Jurgen Bohn. Wireless internet access - security guide, June 2001.
- [SIR01] Adam Stubblefield, John Ioannidis, and Aviel D. Rubin. Using the Fluhrer, Mantin, and Shamir Attack to Break WEP. Technical Report TD-4ZCPZZ, AT&T Labs, 2001.
- [Sti95] Douglas R. Stinson. *Cryptography Theory and Practice*. CRC Press, 1995.
- [Usk97] Sami Uskela. Security in wireless local area networks, 1997.
- [Ven01] R. Venkateswaran. Virtual private networks. *IEEE Potentials*, 20(1):11–15, Feb-Mar 2001.
- [Vic02] Pete Vickers. Sockets project with eVB and VB6. www.devbuzz.com/content/zinc-evb-sockets-pg1.asp, May 2002.
- [Vin01] Joel M. Vincent. iPAQ H3100/H3600/H3700 Series Pocket PC Battery, October 2001.
- [Wea00] Sultan Weatherspoon. Overview of IEEE 802.11b security. *Intel Technology Journal*, 2nd Quarter 2000.
- [WIASG01] Air Force Research Laboratory Wireless Information Assurance Steering Group. Point paper on information assurance for wireless local area networks. Internal Report, May 2001.
- [YF00] Bin Yoa and W.K. Fuchs. Proxy-based recovery for applications on wireless hand-held devices. *The 19th IEEE Symposium on Reliable Distributed Systems*, pages 2–10, October 2000.
- [You00] Roger Younglove. Virtual private networks - how they work. *Computing and Control Engineering Journal*, pages 260–262, December 2000.

Vita

Captain John Lee Camp began his undergraduate studies at the University of Florida in August 1994. In May 1998 he graduated with a Bachelors of Science in Mathematics and was commissioned in the Air Force through the ROTC program.

John attended the initial offering of the Air and Space Basic Course at Maxwell AFB in August of 1998. His first duty assignment was the Air Force Research Laboratory, Human Effectiveness Directorate, at Wright Patterson AFB where he served first as a Research Scientist, then as Chief, Modeling and Simulation Technology Development. While stationed at Wright Patterson, John was awarded a Dayton Area Graduate Studies Institute scholarship and began attending AFIT as a part time masters student in 1999. In June 2002, after receiving his degree, John will be moving to the Air Force Academy where he will teach Mathematical Sciences.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 074-0188		
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</p>					
1. REPORT DATE (DD-MM-YYYY) June 2002		2. REPORT TYPE Master's Thesis		3. DATES COVERED (From - To) Jan 2001 - Jun 2002	
4. TITLE AND SUBTITLE PERFORMANCE ANALYSIS OF A SECURE IEEE 802.11B WIRELESS NETWORK INCORPORATING PERSONAL DIGITAL ASSISTANTS			5a. CONTRACT NUMBER		
			5b. GRANT NUMBER		
			5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S) Camp, John L., Captain, USAF			5d. PROJECT NUMBER		
			5e. TASK NUMBER		
			5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 P Street, Building 640 WPAFB OH 45433-7765			8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GCS/ENG/02-10		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFRL/HESS (AFMC) Attn: Dr. Michael J. Young 2698 G Street WPAFB OH 45433-7765			10. SPONSOR/MONITOR'S ACRONYM(S)		
			11. SPONSOR/MONITOR'S REPORT NUMBER(S)		
12. DISTRIBUTION/AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT Research results of this indicate very poor performance of a Wireless Local Area Network (WLAN) utilizing PDA's. Network throughput is adversely effected most by VPN implementation and slightly by increased file size. The client distance factor has virtually no effect on the throughput. The impact of each of these factor levels is small when compared to the magnitude of the overall mean throughput (< 6%). The average network throughput with the PDA client is much lower than expected ($\approx 11,500$ bps). This is attributed to several factors with degradation primarily resulting from limitations of the PDA hardware and O/S. Because of the low throughput values achieved (regardless if VPN is off or on), an operational WLAN with PDAs (as tested) is not feasible. Operational use of the network tested would require an in-depth analysis of the type of network traffic and performance required to maintain functionality. To deploy such a system, custom designed Winsock controls would need to be implemented to minimize limitations imposed by the PDA. As PDA technology continues to develop, future hardware and O/S functionality may provide a more robust platform for network communications. The battery life of the PDA and jacket battery combination is observed to be about 164 minutes with additional jackets adding about 99 minutes each.					
15. SUBJECT TERMS Wireless local area network, personal digital assistant, virtual private network, performance analysis					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Rusty O. Baldwin, Maj USAF (ENG)
U	U	U	UU	150	19b. TELEPHONE NUMBER (Include area code) (937) 255-6565, ext 4612; e-mail: Rusty.Baldwin@afit.edu
					Standard Form 298 (Rev. 8-98) Prescribed by ANSI Std. Z39-18