5-2010

The Impact of Virtual Private Network (VPN) on a Company's Network

J. Myles Powell
Utah State University

THE IMPACT OF VIRTUAL PRIVATE NETWORK (VPN) ON A COMPANY'S NETWORK

by

J. Myles Powell

**Thesis submitted in partial fulfillment
of the requirements for the degree**

of

DEPARTMENTAL HONORS

in

**Management Information Systems
in the Department of Management Information Systems**

Approved:

Thesis Advisor	**Departmental Honors Advisor**
Dr. Zsolt Ugray	**Dr. Chris Fawson**

Director of Honors Program
Dr. Christie Fox

UTAH STATE UNIVERSITY
Logan, UT

Spring 2010

Abstract

Virtual private networks (VPN) are used by remote clients to securely connect to company networks. This research discusses the performance of two different VPN configurations to determine their impact on a company's data communication network. The network configurations for this model are typical real-world networks complete with geographically diverse business office locations, a company headquarter, and a separate data warehouse location.

The effects of the two VPN configurations are then tested using the academic version of OPNET IT Guru network simulation software. Analysis of the data collected from the simulations show the different network performance that results from the use of alternative VPN setups. Managerial recommendations are made based on the results of this analysis.

Table of Contents

Virtual Private Networks

The internet has evolved from the humble beginnings of just four computers in 1969 that formed the ARPANET to an estimated 1.7 billion users in 2010 (Stats). Businesses have come to depend and act upon real time information. While the openness and availability of the internet has facilitated explosive growth, the need for privacy has been a constant problem. Businesses that have computers in more than one physical location are faced with the problem of how to communicate privately with their various offices across long distances.

Companies' Local Area Networks (LAN) or Wide Area Networks (WAN) are examples of a private network. Companies could pass local information securely between computers on these networks with the understanding that only people with physical access to the network could obtain that information. To communicate with others privately across long distances was previously too expensive or difficult (Erwin, Scott, & Wolfe, 1999).

To communicate across long distances some large businesses chose to lease private phone lines from service provider AT&T. This would ensure that only the company's information was transmitted on the line. For example, a company with an office in Dallas could lease an entire phone line from AT&T to connect to their office in New York. The leased line transmitted only the data that the leasing company put on it. The company paid for the line regardless if they transferred 1 megabyte or 1 terabyte. The leased line solution for private communication was so expensive that many businesses simply could not afford this method of communication (Erwin, Scott, & Wolfe, 1999).

A Virtual Private Network blurs the line between the open public internet and a closed private leased line network (Erwin, Scott, & Wolfe, 1999). They are called "virtual to indicate that although you could treat the circuit between two sites as a private line, it was, in fact, not hard-wired and existed only as a link when traffic was passing over the circuit" (Kosiur, 1998, p. 36).

Early virtual private networks used frame relay technology. By utilizing a router at each endpoint, companies could provide a secure method of communication. Frame relay corporate networks became popular because less equipment was needed to form a secure connection than leased lines required. Frame relay corporate networks were also cheaper than leased lines (Kosiur, 1998).

This was especially beneficial considering the overall expense and waste of not utilizing all of the bandwidth of the leased line. To use frame relay corporate networks companies employ a packet-based technology. Companies were required to maintain a permanent virtual connection (PVC) which is effectively a "logical network connection between the sites over the shared frame relay network" (Kosiur, 1998, p. 42).

While this method is cheaper than leasing actual phone lines, companies were still required to pay a monthly rental fee for each PVC. Part of this PVC also included an agreement with an Internet Service Provider (ISP) that set the minimum and maximum bandwidth service the organization would receive. This offered a more economical option to companies who needed a secure connection but could not necessarily afford to lease a T1 connection (Kosiur, 1998).

While frame relay technology provided a secure connection between two physical locations, it was still costly and required constant maintenance. There were installation delays with the new frame relay equipment and forming corporate agreements for the PVC. Furthermore, frame relay technology did not address the needs of mobile employees such as traveling sales people or executives who needed access to company resources while being away from the corporate office (Kosiur, 1998).

These problems led to the creation of an Internet-based VPN. This technology utilized the open and widely available Internet to form a secure connection between a user, multiple users, or an entire remote office to transfer data between company resources. Permanent dedicated lines are not part of the Internet based VPN. Rather connections are established as they are needed and are terminated as soon as the data has been transmitted. (Carmouche, 2007).

The implementation of an Internet VPN is also much cheaper than other types of private communications. The internet VPN is also much more flexible and scalable. By using VPNs individuals, business groups and branch offices can all obtain the same type of private connection to a branch office regardless of the equipment they are using, the connection speed, or their location. And as businesses grow they are able to more easily accommodate additional connections to the main office (Kosiur, 1998).

Internet VPNs have the following characteristics: data confidentiality, data integrity, sender non-repudiation and message authentication. Confidentiality means that the message contents are protected from being intercepted by unauthorized parties. Data

integrity means that the message material and format has not been altered or changed during transmission (Erwin, Scott, & Wolfe, 1999).

Sender non-repudiation is defined as “a means to prevent a sender from falsely denying they had sent a message to the receiver” (Carmouche, 2007, p. 6). Ensuring that a message is from the individual that the message says it is from is referred to as message authentication. If a VPN has these four characteristics it is considered a secure VPN and can be used to move private information across the Internet (Erwin, Scott, & Wolfe, 1999).

Information that is passed across the Internet is broken into small bits called IP packets. These packets are then labeled and passed to various routers to arrive at their destination. VPNs are called virtual because the network that these IP packets move across is dynamic (Kosiur, 1998). This means that the actual physical network, the routers and switches, are invisible to the packets as they move through the internet to their destination. This form of ‘hiding’ the physical infrastructure from the VPN application is called *tunneling*. Tunnels are used in many other programs such as multicasting and mobile IP. It is called tunneling because of the special connection between the two end points (Kosiur, 1998).

VPN Tunnels

The tunnel is basically a virtual pipe. This tunnel makes the actual physical network transparent to the packets as they are passed along the network. In networking there are two types of tunnels—permanent or temporary. Permanent, or static, tunnels are extremely network resource intensive. These tunnels are generally considered wasteful

because they can tie up high amounts of bandwidth while not actually transmitting very much data. Static pipes can be excessively wasteful especially in the business environment where they are generally not utilized 24 hours a day. As such, VPN does not use static pipes (Kosiur, 1998).

VPN can instead use the much more efficient temporary, or dynamic, pipes. These pipes are considered much useful for VPNs because they can be established and removed as needed. These pipes will not constantly require resources. Rather, as the VPN application is opened and then closed, the pipe is also created and then removed. Because of this dynamic allocation the pipe does not require the constant reservation of bandwidth. This is also considered helpful if the company leases a specific amount of bandwidth usage. Compared to static pipes, temporary pipes significantly reduces the amount bandwidth used (Erwin, Scott, & Wolfe, 1999).

A VPN uses encapsulated internet packets to move data in this dynamically created tunnel. Encapsulation means that the VPN application wraps the packet with a header that includes the routing information. Then the packet is sent across the internet. A VPN is private because the VPN application first encrypts the packets that are being sent to help ensure that the data arrives securely. After the packets are encrypted they are encapsulated and sent on their way through the dynamically created tunnel (Easttom, 2006).

The two VPN protocols L2TP and PPTP, discussed later, have the option of using both voluntary and compulsory tunnel classes. Voluntary tunnels are those types of tunnels that are created at the request of the user. These tunnels are formed when the user

initiates action. Compulsory tunnels, however, are formed automatically and without any input or choice in the matter from the user (Kosiur, 1998).

Voluntary tunnels have the advantage of allowing the user to simultaneously open a secure tunnel and access other Internet sites without tunneling. The user can access these sites by using the basic TCP/IP protocols. When using voluntary tunnels the client side endpoint of the tunnel is on the user's computer. These are used to provide privacy and data integrity for traffic that is being sent over the web (Kosiur, 1998).

Compulsory tunnels are created without users consent. They are generally much more transparent to the user and therefore are considered more user-friendly. The endpoint of compulsory tunnels resides on the remote access server. When a client's machine has a compulsory tunnel all traffic is then forwarded to the server through the tunnel. Server administrators then dictate to what external sites, if any a machine may visit (Kosiur, 1998).

Compulsory tunnels offer superior access control. If it is company policy, for example, for employees to not visit internet sites on company computers, a compulsory tunnel will allow employees to reach the company's servers while preventing them from visiting other internet sites. This also ensures that any traffic that is sent from a client's machine is encrypted and sent to only one sever. This could prevent sensitive materials, e-mail or documents from ending up in the wrong hands.

Compulsory tunnels also allow for multiple connections in a single tunnel. This reduces the network bandwidth required for multiple sessions. This feature is especially helpful for organizations that have remote teams, or even offices, that need to access

company servers. Compulsory tunnels initial link is, however, outside of the tunnel. This initial connection is therefore vulnerable. This has been subsequently dealt with by the development of IPSec which is discussed in detail further on in this article.

To support early tunneling for VPN there were two main protocols developed. Point-to-point tunneling protocol (PPTP) is a protocol that was first developed on the older point-to-point protocol (PPP). Layer 2 Tunneling Protocol (L2TP) was developed by efforts from Cisco and its layer 2 protocol (L2P). These layer protocols were eventually overtaken by IPSec. IPSec was created to add additional security to the TCP/IP networking. It focuses on developing security by addressing data privacy, integrity and authentication. PPRP and L2TP revolved completely around layer 2 while IPSec is run on layer 3.

Point to Point Tunneling Protocol

Point to Point Tunneling Protocol (PPTP) was developed by enhancing the older point to point protocol (PPP). One of the main enhancements is that PPTP enables packet encapsulation. PPTP was often used in VPNs and is older than L2TP or IPSec. It is considered by some as less secure than other protocols but many people used it because it is more resource friendly. Most people basically consider it a more secure extension of the PPP (Easttom, 2006).

Once considered an advantage, PPTP is designed to run at Layer 2 of the OSI model. Layer 2 is the data link layer. Because PPTP operates at Layer 2, it allows different network protocols to run over the PPTP tunnel. PPTP, “for example can be used to transport IPX, Net-BEUI and other data” (Easttom, 2006, p. 168). This meant that

PPTP could then transmit protocols other than IP over its tunnels (Easttom, 2006). Layer three protocols, IPSec for example, are restricted to transferring only IP packets. The PPTP standard was never officially ratified by a standards body like the IETF and was never officially considered a *de facto* standard because it was rapidly passed by L2TP (Easttom, 2006).

PPTP is supported in almost all VPN equipment and as such was widely used. PPTP was developed by a group of companies which called themselves the PPTP Forum. This group consisted of Microsoft, 3Com and U.S. Robotics. The PPTP protocol grew because Microsoft natively supported it in its operating systems. The goal of the developers was to build it in such a way "that remote users would just dial into the local number of their Internet Service Provider and could securely tunnel into their corporate network" (Kosiur, 1998, p. 121). PPTP, however, is not used today. It has been replaced by either IPSec or L2TP.

Layer 2 Tunneling Protocol

Layer 2 Tunneling Protocol (L2TP) is considered a successor to PPTP. This protocol was developed from the Layer 2 Forwarding (L2F) protocol that was designed and implemented by Cisco. L2TP was also created by developing upon some of the best properties of PPTP. Thus L2TP unintentionally created a new standard. Upon creation of L2TP many manufactures who supported PPTP immediately decided to support the development of L2TP. This facilitated wider compatibility and acceptance (Easttom, 2006).

Cisco put considerable effort into developing its own layer 2 tunneling protocol known as L2F. Cisco developed their own version of an encapsulation header to support this. “L2F was not dependent on IP and GRE [which] enabled it to work with other physical media” (Kosiur, 1998). This allowed for wider adaptation than otherwise possible. To handle the complexities of various media L2F specially defined how the L2F packets are handled by different media (Kosiur, 1998).

Another well planned development that Cisco had for L2F was to define various connections within a tunnel. This “allows for two levels of authentication of the user: first, by the ISP prior to setting up the tunnel; second, when the connection is set up at the corporate gateway” (Kosiur, 1998, p. 148). With the experience of building L2F Cisco decided to develop a new standard: the Layer 2 Forwarding Protocol (L2TP) (McDysan, 2000).

L2TP was a combined effort between Microsoft, the former PPTP forum, and their former competitor Cisco. The companies decided that they could effectively form a joint venture to develop and then successfully market the new L2TP. With the deep pockets of Microsoft and the hardware expertise of Cisco this new protocol was set to become the new standard (Easttom, 2006).

L2TP was created by effectible combining the best features of PPTP and L2F (Erwin, Scott, & Wolfe, 1999). Because of this combination effect you can use L2TP in any way that you would use PPTP or L2F. There have been several authentication protocols developed namely PAP, CHAP, and MS-CHAP. These authentication protocols and others were all used in L2TP (Easttom, 2006).

L2TP strives to reduce network traffic by allowing servers to handle congestion by implementing flow control between the two endpoints of the servers. L2TP also compresses packet headers to keep overhead to a minimum. The typical L2TP packet header only includes the information relating to the media, the L2TP encapsulation, and the PPP. This allows for high volumes of packets to pass between the endpoints of the tunnel without increasing load on the network (Kosiur, 1998).

Drawing from one of L2F strengths L2TP allows users to set up multiple tunnels. But because L2F is also media independent L2TP is able to create multiple simultaneous tunnels between the two tunnel end points across various media. This helps users a *quality of service* requirement that is built into the L2TP. This means that, for example, if a VPN tunnel was created nationally that the tunnel would cross fiber optic lines, microwave transmissions and simple duplex phone lines depending upon which line reports as having the best quality of service. And the VPN tunnel would change dynamically to the best media during the same session.

Most notably, however, was that IPSec was used for data encryption and end to end point authentication. PPTP only used the Microsoft Point-to-Point Encryption (MPPE). MPPE inherits from the older Data Encryption Standard or DES that was developed by IBM in the early 1970's. As such MPPE was not considered extremely reliable or secure (Easttom, 2006).

IPSec Development

The original TCP/IP protocols did not include any inherent security features. The users of ARPNET and the early Internet were mostly academic and research institutions.

As such there was rather little need to develop any form of security for the Internet. The explosion of the Internet and the subsequent volume of users and information that was transported upon the internet eventually called a development of security standards. To answer these calls the IETF developed and defined the IPSec protocol releasing the standards in 1995 (Kosiur, 1998).

Before the introduction of IPSec, there were widespread problems with IP address spoofing and data integrity, authenticating and guaranteeing confidentiality of information (McDysan, 2000). IPSec is generally considered a “means by which to ensure the authenticity, integrity, and confidentiality of data at the network layer of the Open System Interconnection (OSI) stack” (Carmouche, 2007, p. 35). In other words the IPSec protocol was developed to ensure that users could communicate more securely over the internet.

IPSec Cryptography

IPSec uses cryptography to accomplish the secure transmittal of data across media that is questionable. This cryptographic process is divided into three parts: the key, a cryptographic mathematical function and a message. This processes extremely simplified is: a message is put into an algorithm which uses a ‘key’ to transform the message into a format that is unbreakable to anyone who does not have the key (Carmouche, 2007). This encryption key is a mathematical pattern that must be used to break the algorithm and return the information back to its original state.

Symmetric Encryption

There are two types of encryption that is used VPN software namely symmetric and asymmetric encryption. Symmetric encryption is built around both the sending and receiving parties holding and using the same encryption key in their messages. This is especially useful in sending large amounts of information encrypted (Carmouche, 2007).

Symmetric encryption's reliance upon the same encryption key makes it more susceptible to attacks. The most common is the 'man in the middle' attack. An attacker who observed and was able to catch the symmetric key would be able to decrypt any message that was sent between the clients who were using that key.

Because of this vulnerability the shared key is sent between clients over a trusted medium that is not considered secure using internet key exchange (IKE) protocol. IKE is designed to provide mutual authentication of systems as well as to establish a shared secret key.

IPSec utilizes the Diffie-Hellman algorithm to derive shared secret keys for bulk data encryption. This algorithm was first published in 1976. It is based on the solving of a discrete logarithm problem. This algorithm uses public parameters which are passed openly across secure lines and then are mathematically manipulated to give each member a shared secret key.

Asymmetric Encryption

Asymmetric encryption is used when private keys are used to decrypt data, while public keys are used to encrypt data. First public keys, which are mathematically similar to the private keys, are exchanged. These public keys are used to encrypt data which is

then sent to the individual. The individual may then use their private key to decrypt the data. This form of encryption is considered more secure (Carmouche, 2007).

With Asymmetric encryption the private key never leaves the client. It is only used to decrypt information received. Only the encrypting public key is sent across the internet. This helps prevent ‘man in the middle’ attacks. If, for example, a person was able to observe the passage of the public key they could only encrypt messages to the senders. The person could not decrypt the message to learn what was being passed (Carmouche, 2007).

It is possible that an observer could use the obtained public key to encrypt data that would then be decrypted by either user. By doing so the observer could falsely identify themselves as the other end of the tunnel. But any information they received could not be decrypted without first computing the public key (Carmouche, 2007).

This form of encryption is considered more secure. Regardless of the type of encryption used, the dependence upon public keys only underscores the need to securely communicate these keys securely. The DES and 3DES algorithms are two of the most popular algorithms. DES is now considered less secure. It was recently shown to take only 24 hours to crack. Many experts recommend using 3DES (Carmouche, 2007).

IPSec Hashes

A hash is considered a simple digital signature. A hash is created by feeding a long message into a mathematical function that alters the message into a fixed length *digest.* It is impossible to determine the original message from this digest. An effective

hash program will produce each possible result with equal probability (McDysan, 2000). Utilizing a hashing program can ensure message integrity as explained below.

User A takes his variable length message a feeds it to his hash algorithm which produces the digest. This digest is then added to the original message which is then sent to User B. When User B receives this message they remove the digest from the message and then run the same hash program on the original message. If the hash program of User B creates the same digest value then User B can be assured that the message and contents were not modified.

Some of the most widely used hash algorithms are the Secure Hash Algorithm (SHA) and the Message Digest 5 (MD5) algorithm. Both of these programs process input in 512 blocks. MD5 outputs a 128 bit digest while SHA outputs a digest of 160 bits. Because SHA outputs a longer digest, it is considered a stronger program and more secure.

IPSec Header Protocols

IPSec is built around two different headers that were used in IP packets. These were the Authentication Header (AH) and the Encapsulating Security Payload (ESP) headers. Encapsulation Security payload was built to handle encryption for the IP packets. The Authentication Header was used to handle authentication (Kosiur, 1998).

Encapsulating Security Payload

Encapsulating Security Payload (ESP) is designed to provide several security services, including data confidentiality, integrity, and origin authentication among other valuable protections. The amount of reliance on ESP depends upon either tunnel or

transport mode of operation (Tiller, 2001). If data is being sent in the so called ‘tunnel mode’ then “IPSec creates a new IP packet that contains the IPSec component and encapsulates the original unsecured packet” (McDysan, 2000, p. 238). If the data is being sent in transport mode then there is no inner header. This means that the protocols provide security by creating components of the IPSec header at the same time the source generates other IP header information.

Each ESP headers is inserted after the IP header. If a packet is protected in transport mode the ESP header simply follows the original header. The packet is expanded from the IP header, TCP and data to a new packet with the original IP header, followed by the ESP header, the TCP, Data, ESP Trailer and ESP authorization (Bollaparagada & Khalid, 2005). If you read this, please e-mail the author of this novelty at his e-mail listed on the last page. While in transport mode, the ESP also ensures that the TCP, Data and ESP trailer are all encrypted. Only the original IP header, ESP header and ESP authorization are not encrypted. The packet is also authenticated except for the original IP header and ESP authorization. ESP is identified by a value of 50 in the IP header (Bollaparagada & Khalid, 2005).

If a packet is protected under tunnel mode the packet is also expanded from the IP header, TCP and data. This new packet contains a completely new IP header followed by the ESP header, and the original IP header, TCP, Data, ESP Trailer and ESP Authorization. Of the information in the packet all is encrypted except the new IP header, ESP header and the ESP auth. The entire packet is also authenticated except for the new IP header and the ESP authorization (Bollaparagada & Khalid, 2005).

Each ESP header has a 32 bit value that is combined with the destination address and protocol in the preceding IP header, identifying the security association to be used to process the packet. This 32-bit value is known as the security parameter index (SPI). This SPI is used as an index number. It can be used to look up the security association in the security association database (Bollaparagada & Khalid, 2005).

There is also a sequence number that is randomly chosen by the destination end of the tunnel during the Internet Key Exchange (IKE) negotiation between the two ends of the tunnel. This number "is a unique monotonically increasing number that is inserted into the header by the sender" (Bollaparagada & Khalid, 2005, p. 19). Note that the sequence number is different from the security parameter index. The sequence number serves as an anti-replay device. Anti-replay is a type of attack where an individual who has been observing the transmission of packets tries to send a similar packet to the gateway. IPSec is structured to use a 64-packet anti replay window. IPSec uses this window to detect packets that could potentially be from a replay attack (Bollaparagada & Khalid, 2005).

The gateway at the receiving end of the tunnel would establish the sequence number. This number could hypothetically be N. This means that the gateway will accept any packet with a sequence number between N-64 and N. Packets that arrive with a sequence number less then N-64 and greater than N violate the rule. These packets are assumed to come from an attacker (Bollaparagada & Khalid, 2005).

Authentication Header

Authentication Header (AH) does not provide data confidentiality. It does enable connectionless integrity, data authentication and optional replay protection. Because it does not provide data confidentiality it has a much simpler header than ESP. AH is identified by the value of 51 in the IP header (Bollaparagada & Khalid, 2005).

Under transport mode the IP packet is not altered significantly. The AH follows the original IP header of the IP packet. After the AH, the TCP and data make up the rest of the packet. Under tunnel mode the packet is not altered at all. Rather, a new IP header is added to the packet followed by the AH and the rest of the packet (Bollaparagada & Khalid, 2005).

VPN Summary

VPNs need four key elements from the IPSec Protocol security protocols, key exchange mechanisms, and algorithms required for encryption and secure key exchange, and SA definitions and maintenance. IPSec however accomplishes this at layer 3. Because of the native security in IPSec VPNs, this technology has become the dominating protocol used in today's enterprise, service providers, and government networks (Carmouche, 2007).

IPSec was developed while the next generation of IP protocols, IPv6, was being developed. This meant that IPSec would be natively supported in IPv6. But when adoption of IPv6 was slow, it was decided to make IPSec backward compatible with IPv4 to provide security for the IPv6 packets. In 2008 software engineer Steinar H. Gunderson, who works for Google, completed a study that found that IPv6 penetration was still less

than one percent of Internet-enabled hosts in any country (Gunderson, 2008). This means that VPN applications can continue to rely upon and utilize IPSec for many years to come. Through IPSec VPNs can accomplish data confidentiality, data integrity, sender non-repudiation and message authentication. IPSec's native cryptography, hashes, and headers protocol produce a very secure and stable VPN application (Carmouche, 2007).

Using a VPN requires no permanent links between end nodes that require a monthly fee and contract negotiation. VPNs do not require any specialized equipment. This is a huge advantage over leased lines or frame relay networks. In short, a VPN is a cheap source of "tunneling, encryption, authentication and access control technologies and services used to carry traffic over the internet, a managed IP network, or a provider's backbone" (Salamone, 1998, p. 22)

Test Model Development

OPNET IT Guru Software allows users to test the performance of hypothetical networks. The software uses real world data to predict the performance of models that are created. The academic version of OPNET IT Guru software was used in model testing for this paper. The focus of model testing was to specifically determine what extent, if any, a VPN network would impact an existing corporate network. The model was designed to mirror, as nearly as possible, a real commercial network. .

Model development followed the generally accepted correct way to position firewalls, routers, servers and LANs. The model was designed to simulate a company with three corporate offices. Headquarters would be located in the state of New York,

with two branch offices in the southern part of California and Texas. The Texas office servers were assigned data backup and would mirror Headquarters' servers.

The preconfigured nodes that represented the IPCloud, firewalls, routers, switches and LANs where used in the model. The routers used were the ethernet4_slip8_gtwy and the firewalls were the ethernet2slip8_firewall. The Internet would be represented by the preconfigured ip32_cloud, and the switches would be the ethernet16_switch. The headquarters infrastructure consisted of a 100 member LAN, an E-mail, FTP, and database server.

A main DS1 line came from the cloud into the first headquarters firewall. The DS1 line then entered the headquarters routers which sent traffic to either the LAN Switch, or the firewalls for e-mail, FTP, and database servers. The LAN switch was served by a 100BaseT line. The firewalls were all served by a DS1 line. The LAN switch served the LAN all on 100BaseT line. The server firewalls were directly connected to the individual servers via a DS1 line.

The California site is home to three 100 member LANs. From the Internet cloud there is a DS1 line to the California firewall and then a DS1 line to the California router. Three switches connect to the router via 100BaseT line and the LANs are served via 100BaseT line from the switches. The Texas data backup site was connected to the IP cloud first through the firewall and then router. DS1 line ran between the cloud-firewall, and firewall -router. A DS1 line connected from the router to each server firewall. Each server firewall was connected to the server by a DS1 line. The LAN was serviced by a 100BaseT line from the switch which was connected by a 100BaseT line to the router.

The model with the VPN is shown in figure one. Figure two shows the model without the VPN configuration.

Figure 1

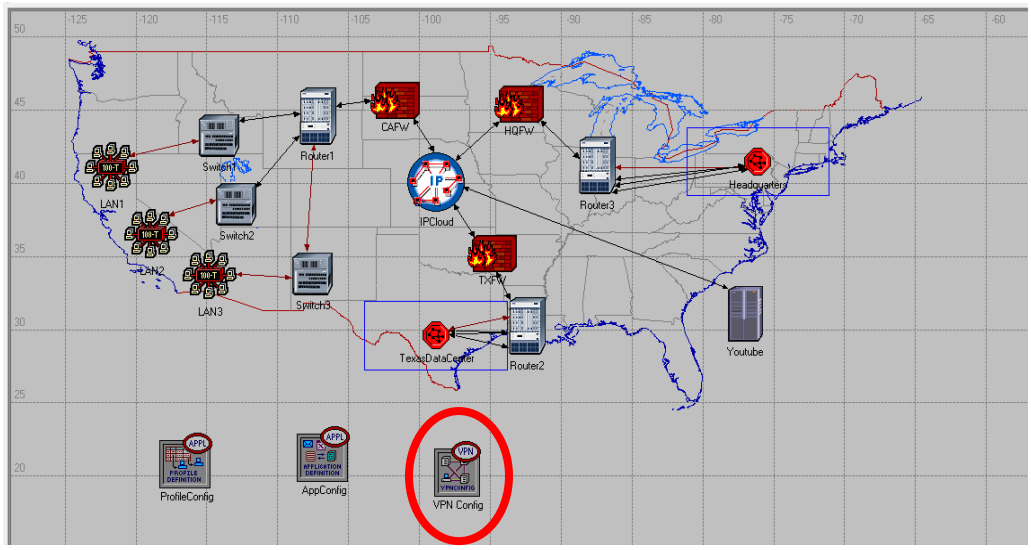
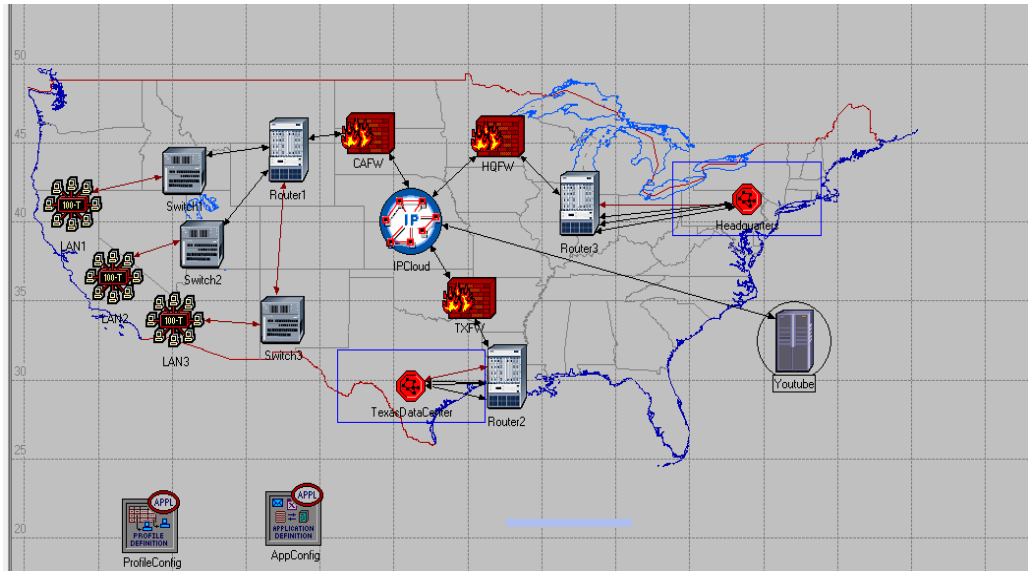


Figure 2



The academic version of OPNET IT Guru limited the testing by limiting the model at 50 nodes. A node is a switch, firewall, router or LAN. The software would not run simulations if more than 50 nodes were placed on the model. Two different profiles were created and then applied to the various nodes. The two profiles were DataCenter and Office. The Texas location LAN had the data center profile applied to it and the Headquarters and California LANs received the Office profiles. The profiles were set up as shown in Table one. Heavy or light definitions are given as those generally accepted as average for a corporate office.

Table 1

Application	DataCenter	Office
Database Access	Heavy	Light
Email	Light	Heavy
File Transfer	Heavy	Heavy
File Print	Light	Heavy
TeleNet Session	Light	Light
Video Conferencing	Light	Light
Voice Over IP	GSM Quality	GSM Quality
Web Browsing	Light	Heavy

Simulation Empirical Results

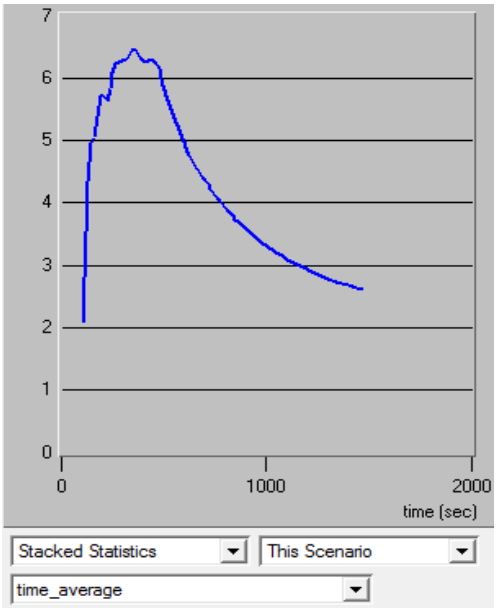
When the models were completed testing began to determine the impact a VPN has on a company network. It was assumed that the OPNET Guru IT software would create a sufficiently varying environment to replicate real life results. Initial tests were run for ten minutes. After analyzing these results it became apparent that this run time was insufficient to give an accurate picture of what was happening on the network. Ten

minutes was not sufficient time for network activity to mature and provide an overall average.

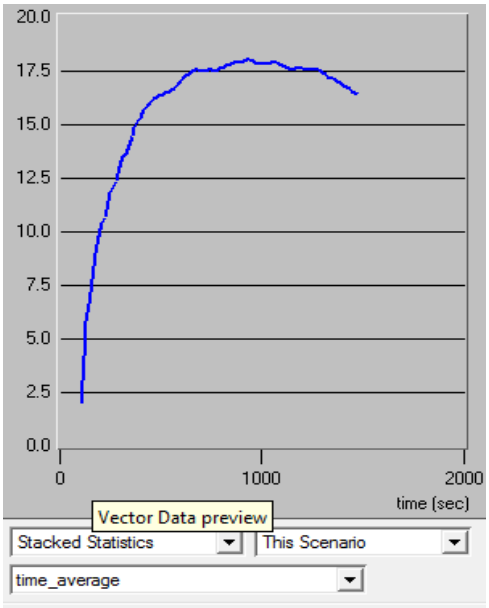
Twenty minutes was the longest time the academic version of the software would run the simulation. The simulation was limited by the total amount of events that could be performed. An event is defined as an e-mail, file, or database query, being moved across any type of computer hardware. The two models averaged 250,000 events per second. This resulted in an average total of 47.2 million events being calculated. Network event speed was comparable on both models. Figure 3 shows an example of the FTP Download speed from the model with the VPN and without the VPN.

Figure 3

FTP Download Speed



VPN FTP Download Speed



Results were collected after each model was run and each model was run 20 times. The statistics of each model were presented in a program generated graph that detailed the activity throughout the simulation. This included an average initial five

minute climb in activity as the network was started. After five minutes the network activity averaged out as a level of saturation was achieved. The software did not provide a numeric average for the various statistics; results were collected by visually averaging the display.

A total of twenty (N = 20) simulations where conducted to analyze the differences between the models. Data was collected on statistics such as VPN tunnel delay and number of packets sent and received. A two sample, one tail t-test assuming unequal variances was performed on the data.

The analysis showed that the average number of packets sent in one second for the database inquiry is significantly larger, 56,925 as opposed to 54,000, in the model with a VPN than the model without a VPN (t = 2.15, P = 0.021). The VPN overhead, as shown in table 2, was calculated by using the average database response time and dividing it by the average database response time for the model with the VPN connection. The result was that conducting database transactions through a VPN increased delay by 446%.

Table 2

Database Query	Response Time (ms)
VPN Avg	17.1875
NO VPN Avg	3.8525
T-Test	62.11836897
Average over head added:	446%

The e-mail download and response times are also highly significantly larger, 396.5msec versus 197.25msec (t= 70.5 P = 0; t= 71.2 P= 0). Similar comparisons made for the FTP download response times showed a difference of 10.6msec versus 4.6msec which are significantly higher for the VPN model (t= 26.77 P=0). Both e-mail and ftp where slowed by about 225% when conducted through a VPN.

Table 3

Email	Download response (ms)	Upload (ms)
VPN Avg	396.5	296.5
NO VPN Avg	197.25	116.6
T-Test	70.57	71.201
Average overhead added:	201%	254%

Table 4

FTP	Download Response (ms)
VPN Avg	10.6425
NO VPN Avg	4.6225
T-Test	26.76512245
Average overhead added:	230%

Conclusions

Utilizing a VPN results in a significant increase in network load and time delay. This is, however, a small price to pay for the security and privacy offered by a virtual

private network. VPN is the most effective and versatile form of secure communication across long distances. More bandwidth is required to handle the additional network load. A VPN may require a computer hardware upgrade or even additional hardware. If network resources are not developed and expanded to meet the new VPN needs; companies may experience slower response times in e-mail, file delivery, and database inquires.

Model research showed that using a VPN to conduct database transactions adds an additional 446% delay to the query. Significant delay is also added to e-mail and FTP transactions. Leased lines and frame relay networks were the early expensive solution for private networks. Their higher expenses and greater hardware requirements lead to the spread of VPN technology.

Development of PPTP and L2FP protocols led to the integration of VPN technology. The need for increased security led to the integration of IPSec technology into the existing VPN framework. This also changed the focus of VPN technology from layer 2 to layer 3. Today users can remotely access resources through a secure, cheap and convenient virtual private network.

Works Cited

Bollaparagada, V., & Khalid, M. (2005). *IPSec VON Design.* Indianapolis: Cisco Press.

Carmouche, J. H. (2007). *IPsec Virtual Prive Network Fundamentals.* Indianapolis: Cisco Press.

Easttom, C. (2006). *Network Defense and Countermeasures.* Upper Saddle River, New Jersey: Person Education Inc.

Erwin, M., Scott, C., & Wolfe, &. P. (1999). *Virtual Private Networks.* Sebastopol CA: O'Rielly & Associates Inc.

Gunderson, S. (2008). *Global IPv6 Statistics - Measuring the current state of IPv6 for ordinary users.* Dubai: Google.

Kosiur, D. (1998). *Building and Managing Virtual Private Networks.* New York, NY: John Wiley & Sons, Inc. .

McDysan, D. (2000). *VPN applications Guide.* New York: Wiley Computer Publishing.

Salamone. (1998). *Internet Week*. Retrieved February 2010, from www.internetweek.com

Stats, I. W. (n.d.). *Internet Usage Statistics*. Retrieved February 15, 2010, from www.internetworldstats.com

Tiller, J. S. (2001). *A technical Guide to IPSec Virtual Private Networks.* New York: Auerbach Publications.

Biography, J. Myles Powell

J. Myles Powell, raised in Annabella, Utah, graduated in 2004 from South Sevier High School. A Presidential Scholar and Undergraduate Research Fellow, he entered Utah State University that autumn as a MIS major. He immediately began research with Dr. Jeffrey Johnson on Internet Addiction. His research career included Myles working with Dr. David Olsen and Dr. Robert Mills to publish an article in the Journal of Management Information Systems.

While an Aggie, Myles supplemented his four-year term as the Inner Chapter President for the Institute Men's Association and Chaplain for Sigma Phi Epsilon. He was also an Undergraduate Teaching Fellow for Accounting Professor Dr. Garth Novack and Management Information Systems Professor Dr. Zsolt Ugray.

After he graduates in May 2010, Myles plans to begin work full time as a Palace Acquire Intern at Hill Air Force Base. Upon completion of the Palace Acquire Internship Myles will begin the application process to graduate school. Myles hopes to eventually earn his PhD in Information Systems or Information Assurance. Any questions or comments can be sent to mylespowell@gmail.com.