Superman

BSc (Hons) Networking and Systems Support

Matric No: S09xxxxx

Module Leader: Richard Foley

Final Honours Report

An investigation into the performance aspects of "traditional" PPTP and modern VPN tunnelling techniques based on small business network architecture.

Project Supervisor: Richard Foley

Second Marker: Tom Buggy

"Except where explicitly stated all work in this document is my own."

Signed: _____ Date: _____

Abstract

Virtual Private Networking (VPN) is a flourishing technology implemented in numerous organisations fulfilling the role of remote access into private company Local Area Networks. Historically reserved for deployment only in the largest of organisations due to the enormous investment required, VPN technology is now widely-available to businesses of all proportions thanks in part to various open-source solutions and Windows Servers' wizard-based implementation.

Organisations are now using the ubiquitous nature of the Internet in conjunction with Point-to-Point Tunnelling protocols (PPTP) and VPNs to securely connect to remote sites. However, small businesses (which make up the smallest percentage of active VPN deployments) are worst affected by relatively expensive business internet connections, usually with speeds only reaching a few Megabit per second. With small businesses recognising the possibilities and improvements that can be achieved by deploying VPNs, they are apprehensive due to relatively slow internet connections. Overutilization of this resource can prove disastrous as many businesses rely on the internet for day-to-day operations.

This project aimed to investigate the performance and efficiency of PPTP and SSL VPN tunnelling techniques when applied in a small business locale. This was achieved through the implementation of realistic lab experimentations designed to simulate the physical layout, and traffic patterns of a typical small business. Whilst PPTP only has one cipher, multiple variants of SSL VPN ciphers were tested and their performances were recorded. The results collected allowed patterns to be identified, as well all results being critically analysed and tested against initial hypotheses.

The results of this study clearly indicate PPTP is the most efficient and highest performing VPN tunnelling method available. However, SSL VPNs were found to only incur a 9% performance drop when implementing the largest and most robust cipher in use – AES 256-bit. This leaves the final decision of which VPN method to use up to organisation or individual preference; do they prefer maximum performance over security, or is a 9% performance drop adequate enough to constitute deployed the strong cipher available in SSL VPNs? In conclusion, it is the author's verdict that this performance drop is acceptable for the increased security it provides. Thus, SSL VPNs can be a cost-efficient and well-performing VPN alternative to traditional PPTP tunnelling methods.

Contents

Chapter 1

Introduction

1.1 Background

Virtual Private Networks (VPNs) are slowly becoming a common sight in organisations today, both large and small, largely due to the incredible developments of the Internet throughout the last 15 years. However, before such advancements organisations relied on dedicated leased-lines to connect remote sites together (Feilner, M. 2006) - an extremely costly endeavour due to the work required to lay a single cable directly between sites. Now any organisation can deploy VPN technologies relatively easily to provide secure remote access to employees when working on the road or at home. Internet Service Providers (ISP) VPN solutions commonly make use of Quality of Service (QoS) to mimic the properties of a dedicated leased-line; ensuring organisations receive priority and enforcing guaranteed speeds based on contractual agreements (Keng Lim, L. et al. 2001). VPN implementations are now being recommended by Network Security consultants over traditional dial-in services as described by (Harmening, J. et al. 2009).

The slow uptake of VPN solutions may be attributed to many organisations that refused to invest due to the potential threats of spoofing and Trojan horses (Hancock, B., 1999.). VPN usage in enterprises was discussed by (Anon, 2004) showing a huge estimated increase of VPN deployments by 2007 - over double the figures from 5 years previous. Although no official figures exist to confirm or contradict this estimation, it can only be assumed that with the introduction of wizard-based VPN deployments on Microsoft Server operating systems and open-source VPN software, that the figures of 55%, 74%, and 90% (Anon, 2004) for small, medium and large enterprises respectively, hold some truth.

However, with VPN deployments becoming simplified it can create significant security ramifications if setup by an inexperienced or unqualified individual - as (Strayer, 2004) states, the "private" in Virtual Private Network is often overlooked. Configuring a VPN deployment must be thought out methodically and with great care. It is not only the term "private" which is being overlooked, so are many other vital steps required to deploy an efficient and effective VPN. According to (Broderick, 2001), it is often organisations that do not generate an analytical and logical approach who fail to deploy a successful VPN solution.

With organisations demanding lower costs and turning to VPN deployments based over the public Internet, they also required the data being transmitted to be encrypted, thus, avoiding any potential breach of sensitive company material. As (Hunt, 2001) discusses, there are

3 essential aspects which must be ensured: confidentiality; integrity; availability. Yet, early Windows implementations of their Point-to-Point Tunnelling (PPTP) encryption quickly became obsolete and broken using relatively simple techniques as discussed by (Munro, 2006). Furthermore, initial implementations of authentication such as PAP (Password Authentication Protocol) had extravagant flaws - the most evident being it sends authentication details using plaintext (Wright, 2000). This would be considered a rather large (yet possibly acceptable) flaw when employed on a secure Local Area Network (LAN) environment, but when coupled with the public internet, leaves all remote user credentials vulnerable to prying eyes. PPTP VPN tunnels are still used consistently throughout many business today due to its simplistic setup and minimalistic configuration, and it what is referred to as a "traditional" VPN.

Later versions of Microsoft VPNs and more recently, open source variants such as OpenVPN, employ much more cryptographically secure encryption techniques. Nevertheless, even the older variations of encryption could encrypt sections of the protocol stack used as well as the payload data (Hancock, 1997).

As mentioned earlier, VPNs are typically being deployed to provide remote access for road-warriors (i.e. sales and marketing employees) and employees working from home. This is extremely beneficial as it provides the ability to access private data as if the user were connected to the organisations LAN (Anderson, 2000).

The latest and most touted VPN solution is the SSL VPN which utilises the Secure Sockets Layer (SSL) protocol found on all web browsers. The ability to deliver remote access to thousands of users in a relatively short amount of time is a key selling point of the SSL VPN - especially useful for deployment in an extremely large organisation. As (Harding, 2003) describes, this VPN solution is perfect for large-scale deployments due to its simplicity, cost-efficiency and time-efficiency. As with most organisations, it can be assumed that the majority of decisions are made on cost-efficiency and return on investment probabilities, and the conclusion by (Forte, 2009) suggests that SSL VPNs will soon become the de facto standard due to its security and return on investment properties.

Nevertheless, as encryption key sizes increase and mathematical formulae become ever more complex, this causes packet overhead sizes to balloon. With ballooning packet overheads reducing the efficiency and the space available in packets that would otherwise be used to transfer more data, it would be reasonable to theorise that it could cause some sort of negative effect of the performance of a network (Jain, 2002) – more specifically, the throughput of the VPN tunnel.

Similar studies have been produced under simulation conditions; such as the real-time simulation infrastructure based on OpenVPN (Liu et al. 2009). However, as this study was developed using simulations and focuses specifically on OpenVPN performance, there are no other results from any other VPN solution to compare performance metrics against. To complete a comprehensive and well-rounded project of VPN performance aspects, at least 2 solutions will be required to construct an adequate conclusion.

With VPN solutions becoming ever more popular due to increasing simplicity and availability, small businesses are starting to consider VPN deployments over their infrastructure. The ability to allow remote access to private company data from outside the office and to work from home, are huge incentives small businesses envisage. Allowing employees to undertake work from the comfort of their home could improve production and efficiency whilst reducing costs on office leases and equipment purchases.

However, with increasing demand on accessing more data faster, small businesses are reaching the limits of their network architecture. As high-bandwidth leased lines can cost in the region of thousands of pounds per month, small businesses can typically only afford lower speed business grade connections - normally in the region of a few Mb. If organisations were to deploy VPN solutions on a small bandwidth connection, the link could be fully saturated by a handful of remote users, degrading the whole service to a potential standstill. Furthermore, with e-commerce and internet access a necessity and business critical assets for many small businesses, any degradation in performance could prove disastrous.

With every bit of bandwidth available being crucial to the small business operations, the deployment of the most efficient VPN solution is essential. Additionally, to ensure small businesses and organisations of all sizes, as well as many other potential stakeholders, are provided with an essential substantiated and calculated comparison, it is proposed that a study of differing VPN tunnelling methods and their performance metrics is conducted.

As discussed earlier, there are several prominent VPN types available to businesses of all sizes. To provide a reasonable and well balanced study of the performance characteristics of these VPN methods, it would only be logical to select the 2 most popular – in terms of current deployment levels and future trends – VPN methods and complete a thorough analysis of their performance when used in conjunction with a variety of application protocols such as HTTP, etc. Therefore, this project will investigate the performance characteristics and metrics of PPTP (traditional) and SSL VPNs when deployed in a small business network architecture.

1.2 Project Outline & Research Question

This section specifies the project outline and research question that is being investigated, as well as the justification for this area of research. Also discussed is the type of project to be undertaken along with the aim and objectives, and any associated hypotheses.

1.2.1 Research Question

How do differing "traditional" PPTP and current SSL VPN tunnelling techniques affect the performance aspects of data communication based on modern small business network architecture?

1.2.2 Project Type

This project will be an experimental project applying lifelike scenarios on physical hardware, and testing how PPTP and SSL VPN tunnelling techniques affect performance metrics and characteristics when used in conjunction with various application protocols. Results gathered throughout the experimentation will be compared and reviewed for each VPN method and protocol used.

1.2.3 Project Aim

The aim of this project is to investigate the performance characteristics of 2 dissimilar, but popular, VPN tunnelling techniques (PPTP and SSL) when applied in a small business locale. This investigation will make use of physical hardware and accurate network scenarios to obtain authentic results which will be subsequently evaluated based on their performance characteristics. The results gathered at the conclusion of the different scenarios will be scrutinised and compared to determine which VPN tunnelling method is most efficient in modern telecommunications for small businesses.

1.2.4 Project Objectives

The primary objectives of the project are:

- Identify the characteristics and data traffic requirements of modern small business network architectures.

Ascertain a greater understanding of typical small business network designs and traffic patterns to help plan more detailed and relevant project experimentations. Traffic requirements identified here will be replicated in the experiment with the use of a traffic generator. Results gathered on this objective will also reflect the type of test data that will be used in the experiment.

- Identify the underlying technologies used in PPTP and SSL VPN tunnelling.

This will provide a deeper understanding of the core VPN tunnelling concepts as well as the differences between the 2 selected VPN methods. It will detail how the different techniques affect the composition of the packets as they traverse the network, as well as a basis of providing hypotheses on associated performance metrics.

- Identify performance aspects common to both VPN tunnelling methods.

Discover which performance aspects will be best suited for comparison between the tunnelling techniques. This will provide the basis of how to identify which tunnelling technique delivers the greatest performance metrics.

- Devise a realistic small business test network architecture scenario.

The scenario must be relevant and realistic to provide the best possible chance of generating valid results. Without valid results, any conclusions given would only be speculation and not justified by pertinent data.

- Implement realistic small business test network architecture scenario for each VPN method.

The decided upon scenario will be implemented for each VPN tunnelling technique along with the different protocols identified during the data traffic requirement objective. The experiment will return performance results for each different protocol tested. Great care must be taken on deciding how the experimentation is implemented to provide the most accurate data possible, as generating small result sets can result in erroneous data.

- Evaluate results from experimentations

The results must be gathered, analysed and understood to be able to justify conclusions as well as confirm or disprove hypotheses. Conclusions must be drawn to be able to justify which tunnelling method is most apt for small businesses based on performance characteristics results.

1.2.5 Justification

This project seeks to discover the performance characteristics and impacts that 2 dissimilar, but commonly implemented, VPN tunnelling techniques have on different network scenarios based on small businesses. This concept is a booming technology utilised in a multitude of different markets, therefore, this project will possibly be of use to several stakeholders.

VPNs can have many implementations in small businesses: remote access to an internal company file server or intranet; remote access for teleworkers; secure encrypted communication channels between remote sites. Small businesses will be able to analyse the results and conclusions to determine if a VPN deployment is applicable for their organisation. This would be a great resource for them if they are considering implementing a VPN solution into their infrastructure, or wish to upgrade their current VPN solution to a higher performing or more modern tunnelling method.

Medium or enterprise level businesses would also be able to benefit from the results and conclusions created by this project, for similar reasons as small businesses. Enterprise level businesses tend to implement large-scale VPN solutions providing remote access for employees, typically road warriors and teleworkers; however, they are usually implemented using out-dated and proprietary VPN solutions. The results generated from this project would be useful to larger enterprises as it would serve as a documentation of performance characteristics between older and more modern VPN tunnelling technologies, allowing organisations to ascertain if upgrading their current infrastructure would be desirable.

Network administrators and System Administrators will also benefit from the results of this project as it will help them gain a greater understanding of VPN tunnelling principles and characteristics; both old and new. By gaining this knowledge Administrators will be able to utilise the project results as a reference to their own organisations current implementation (or lack thereof). They will then be in a position to ascertain as to whether their organisations solution is adequate and meets their current demands.

1.2.6 Hypotheses

This section will detail any hypotheses that have been formulated based on the collective readings referenced in Appendix 6 and Appendix 7.

- SSL VPN will have reduced performance characteristics on all protocols compared to PPTP due to additional overheads.

From the literature review conducted in Section 2, it was found that when strong encryption is enabled on a VPN, the throughput can degrade to less than 35% of the original link speed and CPU usage peaks at 97% (Evans, 2000). However, due to Moore's law, processing power has drastically increased since 2000, and thus, modern processors can use large-block encryption more effectively (Khanvilkar & Khokhar, 2004).

- Any performance drop on SSL VPN will be minimal (within a 15% margin) compared to PPTP.

Based on findings of performance evaluations performed by (Khanvilkar & Khokhar, 2004) on various open source Linux-based VPN solutions, it states no method is able to sustain a constant throughput greater than 50% of the original link bandwidth – including OpenVPN. However, the individual undertaking this project has previous experience with various VPN solutions and their performance whilst utilising strong encryption. Using dedicated VPN providers such as *VPNTunnel* with a 256-bit AES cipher, a constant throughput of greater than 90% has been recorded when compared against PPTP performance from the same provider.

- SSL VPNs will be able to implement a wide array of robust encryption algorithms whilst still retaining an acceptable level of performance within the theorised 15% margin.

Material researched in Section 2.2.2.1 shows that PPTPs encryption method, MPPE, has been found fundamentally insecure. However, SSL VPNs have the ability to implement a wide variety of encryption ciphers through the use of the well-known OpenSSL library, which includes the AES cipher – the only open cipher approved by the NSA suitable for top secret information.

Businesses today require VPN technologies which are simple to install and configure, as well as being secure and bandwidth efficient (Morgan & Lovering, 2008). SSL VPNs are now being packaged in new versions of the Windows Server Operating System, and can be configured through an intuitive wizard-based installer, thus, easing the stress and possibility of implementing insecure VPN technologies (Ruest & Ruest, 2008).

1.2.7 Report Structure

The section will detail the structure of the remaining sections of this report, specifying the substance of the literature review, methods, results, and summary and conclusions.

1.2.7.1 Literature Review

The literature review will be discussed in section 2, and will detail the areas of investigation required to conduct an appropriate and relevant study. It will begin with an overview of generic research relating to small businesses and VPNs, and then further discusses the specific elements which relate to the previously identified project objectives.

1.2.7.2 Methods

The methods will be discussed in section 3, and will specify the approach used to conduct the primary research. This section will discuss the reasons behind the experiment type, as well as thorough analysis of the intended architecture, configuration, and implementation of the intended experiment.

1.2.7.3 Results

The results in section 4 will detail the outcome of the conducted experiments outlined and detailed in section 3. This will include commentary and critical analysis on each result gathered in order to test the author's hypotheses. The results will be grouped based on their VPN tunnelling method and cipher employed (in the case of SSL VPNs).

1.2.7.4 Summary and Conclusions

The summary and conclusions in section 5 will critically analyse the results collected and displayed in section 4, testing the authors hypotheses and identifying any noticeable patterns or further analysis that could be conducted. The project limitations and further works that could be undertaken are also discussed in this section. Finally, the project results and summations will be concluded reporting the efficacy of the study.

Chapter 2

Literature Review

This section will detail the findings of the key objectives listed in section 1.2.4 that are required to undertake a complete and thorough literature review, the answers of which will be vital for the undertaking and construction of relevant test scenarios for the experiment.

2.1 Small Business Networks

2.1.1 Architectures

With money being a major factor for most small businesses (typically less than 50 users is the globally accepted number), advanced technology is seldom deployed on such networks due to their large implementation costs and specialist resources required to configure and maintain the technology.

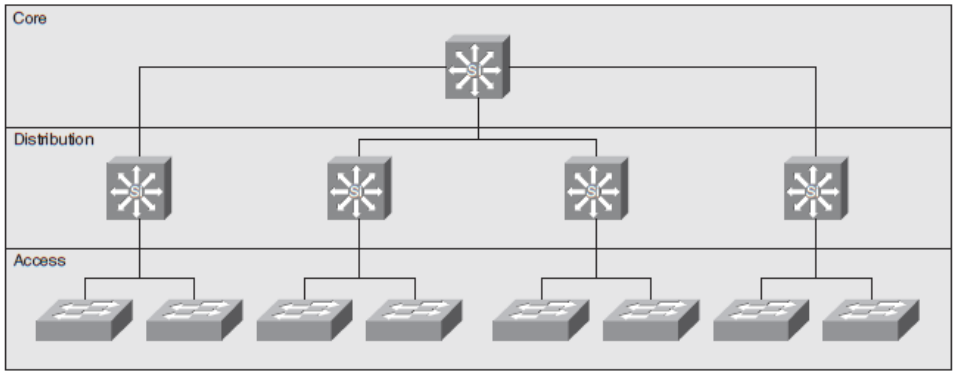


Figure 2.1: Cisco Three-tier Model. Reprinted from "CCNP BCMSN Official Exam Certification Guide," by Hucaby, D., 2007. Copyright 2007 by Cisco Press. Reprinted with permission.

If a small business were to utilise Cisco based equipment to create their network, and follow Cisco best practices of network design, network device prices can range from £2,000 to £50,000, with multiple devices of each type required to provide fault tolerance and redundancy for maximum efficiency. After which, a qualified specialist is required to configure the equipment to the business needs – which can cost as much as £200 per hour. An example of a Cisco recommended

design can be seen in Figure 2.1.

As is evident the costs can mount up rapidly, and large costs are ideally avoided in small businesses. This is why most small businesses are implementing simplistic *star* based network topologies consisting of one or 2 central network devices – specifically unmanaged (basic and requiring no configuration) layer 2 switches. An example of a star topology can be seen in Figure 2.2.

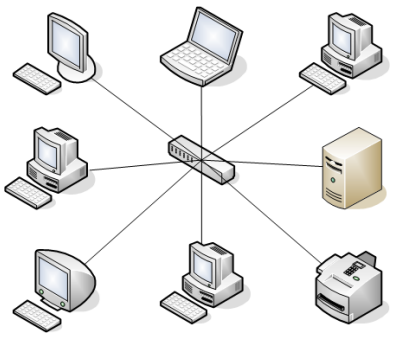


Figure 2.2: Anon. "Star Network Topology". Drawing. *wahyufs.xtgem.com* Unknown <http://wahyufs.xtgem.com/upload/star.png>.

As can be clearly seen, all devices including PCs, printers, laptops, etc. are directly connected to the central switch, providing only a single level of hierarchy. However, even with very little redundancy and fault-tolerance built into the network design, this implementation is proving to be more popular due to the lost costs associated with it. Instead of spending tens of thousands of pounds, companies only require around £1,000 on a single device.

However, with reduced costs comes some serious consequences – the most prominent being the lack of redundancy as discussed earlier in this section. With no redundancy built into the network architecture, the loss of the single switch would result in a complete network-wide blackout. As stated earlier, with e-commerce and internet access a necessity as well as business critical assets for many small businesses (Poon, 2008), any degradation in performance (including Local Area Network and Internet blackouts) could prove disastrous.

Nevertheless, with the overall drive of small businesses to spend as little as possible, important services and necessities are often left without adequate fault-tolerance solely due to the cost involved. This makes the *star* topology the most widely implemented small business network architecture in use today (Gunkel et al., 2008).

2.1.2 Traffic Patterns

An essential part of replicating the conditions of a small business network, aside from its architecture, is to realistically simulate its traffic patterns. Although large organisation tend to utilise large amounts of protocols, both common and intrinsically unique, the majority of organisations today still operate the core 4 protocol types: web, mail, file and domain resolution (Casado, 2007).

With many small businesses now trading online, web access (and thus HTTP and HTTPS protocols) is an essential part of day-to-day activities. Many businesses have started to realise the potential of open source software and the flexibility it can provide them with as stated by

(Fitzgerald, 2006). Being open source allows businesses to tailor ready-made solutions (such as website shopping cart software) with minimal time and effort, resulting in websites being created in days rather than weeks by professional web designers.

To store records of trading activities such as purchases and customer details, databases are commonly implemented as a centralised (and open source – such as MySQL) alternative to commercial (and expensive) implementations. Websites and databases are typically mixed into a single open source solution which businesses of all sizes are utilising. Taking the most popular open source database MySQL as an example, database traffic can be simulated by implementing a database driven website. Using HTTP and HTTPS to call the website script, it will automatically interact with the database generating traffic.

Another common application utilised in businesses of all sizes is e-mail, which is represented by the POP3 and SMTP protocols (ports TCP 110 and TCP 25 respectively). E-mail is an increasingly popular method of providing support to customers for many organisations, as well as inter-office and official business related communications (Byron, 2008).

Some companies also make use of file servers to store large quantities of data for both public and private use. Being almost 40 years old, (Khare, 1998) states that File Transfer Protocol (FTP) is commonly used as it is a mature and stable protocol specifically designed for the distribution of files. FTP traffic occurs over 2 ports (TCP 20 and 21). Port 21 is used for signalling and control messages, whereas port 20 is used for the actual transfer of FTP data.

For all of these protocols to work, another protocol is required to provide translation services – Domain Name Service (DNS). As discussed by (Tyson, 2004), the internet is made up of billions of Internet Protocol (IP) addresses, denoted in the dot-decimal form xxx.xxx.xxx.xxx. As it would be extremely difficult to remember dozens of (up to) 12 digit addresses to favourite websites, they can be replaced for easy to remember alphanumeric addressing. For example to visit the BBC website, instead of remembering the digits *193.105.162.83*, the address *www.bbc.co.uk* can be used. A service to translate the easy to remember alphanumeric address to the dot-decimal format is required so the traffic can reach its intended destination. DNS can be replicated by simulating traffic over TCP and UDP port 53.

2.2 Virtual Private Networks

2.2.1 The Basics

Virtual Private Networks (VPNs) can have multiple functions, but are typically used to provide users or remote sites secure access to internal company networks from external sources – usually via the Internet. As discussed earlier, VPNs were first implemented in the form of direct dedicated links between branches and sites, as this ensured data security and privacy. However, this proved to be an extremely costly endeavour and is not favourable for many businesses today. Therefore other solutions had to be implemented to achieve the same principle, with the most common now being the Internet.

With the advancement of the Internet, it is becoming more commonplace to implement VPNs over the internet utilising encryption to protect sensitive company data (Milanovic, 2001). In recent years, the boom of MPLS VPNs (Multi-Protocol Label Switching Virtual Private

Networks) has meant large organisations can have their traffic and routing information isolated (and secure if employed with encryption) over the Internet, allowing remote branch sites to appear as extensions of their central infrastructure – as if the Internet is an invisible but secure transport medium. However, as with dedicated links, (Uhlig, 2000) states that businesses are being left out due to the cost of renting access to a providers MPLS network – typically ranging in the thousands of pounds per month for high-speed access.

The delegation of responsibility for providing VPN access has consistently been pushed towards the Internet Service Provider (ISP). Originally, it was the organisation that was responsible for implementing the links between external sites, now the ISP controls the majority of the public and private links between sites (Morgan, 2008). With MPLS VPNs, ISPs have also been given the task of securing and isolating individual customer traffic streams which can require expensive equipment – a possible reason for the large price associated with leasing dedicated (virtual or physical) links from ISPs.

There are 2 main types of VPN currently in use today: transport and tunnel. Both methods operate similarly but have one major difference in their implementation. In transport mode the payload of the IP packet is encrypted and/or authenticated. This leaves the original IP header (which includes source and destination addressing) intact and thus open to potential reconnaissance attacks from would-be hackers. In tunnel mode, both the IP header and payload data are encrypted and/or authenticated as can be seen in Figure 2.3. A new IP header is then attached to the tunnelled packet which obfuscates the original IP addressing and replaces it with the newly created VPN tunnel addressing.

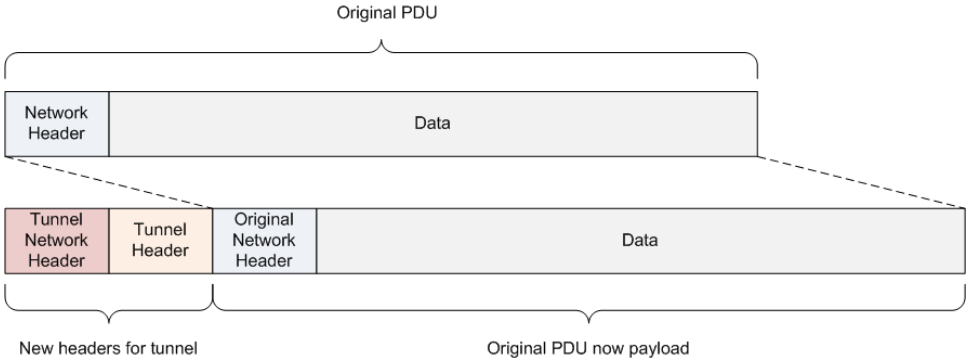


Figure 2.3: VPN tunnel mode packet layout. Drawing. *infrastructureadventures.com*, Keegan, J. <http://infrastructureadventures.files.wordpress.com/2010/12/anatomy-of-a-network-tunnel.png>.

As VPN tunnelling provides increased security by encrypting and/or authenticating a larger amount of the packet, it will be the preferred choice for the experimentation.

2.2.2 Common Implementations

There are a number of different solutions which provide VPN tunnelling, with traditional implementations being Point-to-Point Tunnelling Protocol and Internet Protocol Security (IPsec), and modern variations making use of Secure Sockets Layer (SSL) such as OpenVPN. This section will discuss the 2 most popular implementations and compare their operations.

2.2.2.1 Point-to-Point Tunnelling Protocol

Point-to-Point Tunnelling Protocol (PPTP) is one of the oldest and most widely known implementation of VPNs due to its distribution in all flavours of the Windows Operating System (Berger, 2006). The PPTP specification was first published in 1999 and describes the utilisation of a Generic Routing Encapsulation (GRE) tunnel to encapsulate Pont-to-Point Protocol (PPP) packets.

GRE is a Cisco proprietary protocol designed to encapsulate different network layer protocols inside IP tunnels as described by (Fraser, 2001). This was commonly implemented back in the 1980s and 1990s when Novell's NetWare was a popular Operating System, and networks were required to run a mix of network layer protocols such as IP and IPX (Internetwork Packet Exchange). However, with the advancement and overall adoption of IP, multi-protocol networks died out and GRE's usage has been limited to use in PPTP and IPsec VPNs.

PPTP is typically implemented in a client-server architecture with dedicated hardware to running a server instance of the PPTP protocol. Windows Server is a common platform that remote access is configured on due to its simplicity. Windows Server provides a simple and intuitive wizard-based installation of remote access so that it can be deployed in minutes with minimal configuration. Organisations of all sizes frequently use Windows Server for many aspects of their application infrastructure, with versions of the Operating System even designed specifically for small businesses. Windows Server provides a central point of authentication (Active Directory) as well as many other features such as web server, mail server, file server, DNS server, etc. As a *all-in-one* suite of applications, it holds the largest market share of server Operating Systems with just over 70% (Foley, 2010). Note: when discussing PPTP, the Microsoft PPTP implementation is always assumed.

As previously discussed, both tunnel and transport methods of VPNs introduce overheads to the packets they transmit, PPTP is no different. (Schneier, 1998) describes how the entire IP datagram is encapsulated inside a PPP frame, then the newly created PPP frame is wrapped with a GRE header and an IP header (as can be seen in Figure 2.4). This results in a minimum overhead of 24 bytes (4 bytes for GRE without any options – maximum of additional 12 bytes, and 20 bytes for the new IP header). The new IP header contains routing information relating to the VPN client and the VPN server addressing – obfuscating the original source and destination addressing inside the PPP frame.

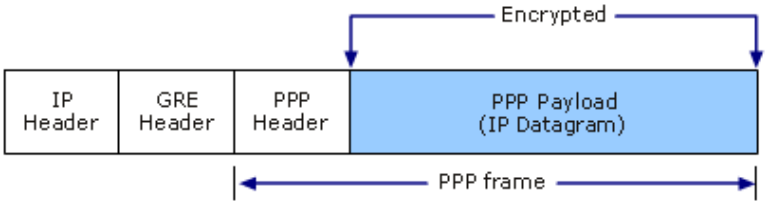


Figure 2.4: PPTP Header Composition. Drawing. *technet.microsoft.com*, Microsoft. <http://i.technet.microsoft.com/dd469817.426583bc-5015-4852-99d9-41a0568262ca(en-us,WS.10).gif>.

With the increased overhead, if an interface is incorrectly configured, a packet may become larger than the interface maximum transmission unit (MTU) permits. The device must then fragment the packet into more manageable pieces, a process which results in more CPU cycles and can affect overall network performance (Gilfeather, 2001).

One major downfall of PPTP is its authentication and encryption capabilities (Schneier, 1998). PPTP has 3 main methods of authentication: PAP, CHAP (Challenge-Handshake Authentication Protocol), and MS-CHAP-v2 (Microsoft Challenge-Handshake Authentication Protocol). PAP transmits unencrypted passwords over the network, thus is considered fundamentally insecure and only used as a last resort. CHAP makes use of a 3-way handshake to verify the identity of remote clients. It is based around the usage of a shared secret that is validated by a *challenge*. After a challenge is initiated, a one-way hash function is used on the shared secret and the *challenge*, after which the returned value is sent back to the challenger for verification. If the values match, the identity of the remote client has been verified. MS-CHAP is similar to the CHAP, except that the plaintext shared secret is not required to be known by both peers. However, MS-CHAP has been noted as having several flaws that makes it vulnerable to brute-force attacks as described by (Schneier, 1999). These include the ability to passively listen to the challenges and responses made by MS-CHAP. With the aid of an open-source hacker tool *L0phtcrack*, the key can then be derived and communications can be deciphered.

Encryption on PPTP is handled by MPPE (Microsoft Point-to-Point Encryption) only when using MS-CHAP-v1 or v2. MPPE's encryption is based on an RC4 stream cipher; however, as there is no method for authentication of the cipher text stream, it is vulnerable to a bit-flipping attack (Schneier, 1998). Bit-flipping is the process of modifying single bits of a stream in transit to change the output without detection.

Overall, it appears that PPTPs age has caused detrimental effects on its ability to securely authenticate and encrypt traffic. However, due to its simplistic deployment and configuration, organisations are not giving credence to the warnings of potential insecurities, possibly in the belief that it will never affect them, and still deploying relatively weak VPN technologies (Utter, 2003).

2.2.2.2 Secure Sockets Layer

Secure Sockets Layer (SSL) has been around for almost 15 years and is the current security standard used in securing many application communications over the internet – especially web browsing and e-mail (Rescorla, 2001). It does this by utilising symmetric cryptography for encryption and a *Hash-Based Message Authentication Code* (HMAC) for authentication. Note: SSL is the predecessor to TLS; however, when discussing SSL, TLS is implied.

Similar to CHAP, (Rescorla, 2001) describes how SSL initially operates in a handshake procedure to negotiate various parameters used to establish a secure connection between the client and server. First, a client connects to a SSL-enabled server and presents a list of supported encryption ciphers and hash algorithms. The server then replies with its preferred (and most secure) ciphers and hash algorithms, along with its identification in the form of a *digital certificate*. The client then verifies the validity of the servers *digital certificate* with the entity who issued the certificate. After verification, the client encrypts a random number with the servers *public encryption key* that was contained in the *digital certificate*, and sends the result back to the server. The server can then decrypt the value with its previously generated *private key*. Finally, with the random number, both client and server can encrypt and decrypt messages to one another securely.

For authentication, SSL utilises a *Hash-Based Message Authentication Code*. This code acts as a signature to all packets transferred between the client and server and verifies the validity and

integrity a packet. A shared secret key is used in a hash function (normally MD5 or SHA-1) to generate the signature. Any packet that doesn't contain the correct signature can be dropped without processing the entire packet (thus not requiring decryption).

Encryption in SSL VPNs (such as OpenVPN) is provided by OpenSSL, an open source implementation of SSL protocols that include various encryption ciphers and hash algorithms, and has been widely available and maintained for over a decade. OpenSSL also packages the AES cipher, which is the only open cipher approved by the NSA suitable for top secret information (Viega, 2002) – a far step ahead of MPPE for PPTP which has well-known cryptographic flaws as discussed earlier. OpenVPN is not limited to a single encryption cipher like PPTP with MPPE; other available ciphers include Blowfish, DES, Triple DES, RC4 and RC5. Therefore, if a cipher is found to be cryptographically unsafe, it is extremely easy to swap to another cipher. As well as not being limited to a specific cipher, by utilising OpenSSL, OpenVPN also has the ability to include brand new ciphers as they are released, allowing for a greater deal of flexibility.

As well as symmetric encryption ciphers, OpenSSL includes public-key cryptography implementations such as RSA and Diffie-Hellman key exchange. RSA (Rivest, Shamir and Adleman) is the first algorithm used in public-key cryptography that is suitable for both signing and encryption. Its most common implementation today is in SSL-based websites providing secure communication (Rescorla, 2001).

OpenVPN can use several different types of authentication including PKI (Public-key Infrastructure) by distributing digitally-signed certificates, pre-shared secrets, and username/password combination (Guo, 2007). It also has the unique ability to utilise all 3 methods of authentication – a great deal more than PPTP provides. HMACs also provide packet authentication and integrity, which can also be described as a *software firewall*.

SSL VPNs appear to have everything PPTP VPNs don't provide – a cryptographically secure method of authenticating peers and encrypting tunnel data. So why do so many organisations still utilise PPTP as their method for remote access? It is possibly due to the complexities of dated SSL VPN technologies that required a great deal of configuration, including PKI implementations and distributing digital certificates (Rowan, 2007). However, this process can now be easily automated through Windows Server.

As with any method of security, this comes at the prices of packet overhead. A typical OpenVPN packet header is comprised of 41 bytes on the security layer overhead (includes signature, sequence number, etc.), and 28 bytes tunnelling overhead (after applying a new IP and UDP header) as described by (Yonan, 2004). This comes to a total overhead of 69 bytes; an increase of 287.5% over the 24 bytes (minimum) that PPTP produces as noted in section 2.2.2.1.

However, this project intends to detail if the extremely large additional overheads that SSL VPNs produces, as well as the large ciphers employed, greatly affects performance when compared to a traditional VPN tunnelling method such as PPTP.

2.2.3 Performance Aspects

With VPNs typically deployed to provide users and sites secure remote access to internal networks, performance is a key issue that must be explored in detail, especially for small businesses that require remote access to business critical communications (e.g. teleworkers). Teleworkers

are becoming an increasingly attractive alternative for businesses that do not want to purchase or lease high-priced offices (Bingham, 2003).

Both (Khanvilkar, 2007) and (Hall, 2008) implemented related performance analysis on different VPN methods; however, utilised very similar parameters of measurements. The most appropriate (and predefined) performance aspects based on previous analyses were chosen as the basis of this projects performance analysis. The most important measurement in regards to VPN performance is tunnel *throughput* (Evans, 2000). *Throughput* is a measurement of data transferred in a specific time period, most commonly denoted in bits per second. In ideal conditions, the *throughput* of a VPN tunnel would be equal to the *maximum sustained throughput* of the original link; however, as VPN tunnels have additional overheads, there is less space in a packet for data. Therefore, the (data) *throughput* of the VPN tunnel will be less than the *maximum sustained throughput* of the original link. This problem can be alleviated by implementing compression over the tunnel, which in practice could lead to a higher *throughput* rate than the original link could achieve, as more data (i.e. more bits) can be compressed into a single packet. The effects of compression will be investigated as part of the experiment.

Another important measurement is packet loss. There are many reasons why a packet would be dropped as it traverses networks, with some of the most common reasons being link congestion and CPU overload. (Barford, 2008) describes how if congestion occurs on an over-utilised link, or an underpowered router in a network is receiving high amounts of traffic over a link, it is feasible that high CPU-utilisation could cause packets to be *missed* and therefore dropped. The same is also true for servers running VPN software. As software encryption is used (as opposed to hardware encryption), extra CPU cycles are needed to encrypt every single packet that will traverse the VPN tunnel (Evans, 2000). With large-block ciphers such as AES-256, this process can be extremely demanding on the CPU, especially on underpowered CPUs, and eventually lead to over-utilisation and packet loss. To overcome this issue, either a less CPU-intensive cipher is required, or a more powerful CPU. As this experimentation will take place use high-spec dual-core machines, it is anticipated that CPU-utilisation, and therefore packet loss will, be non-existent.

Typical Ethernet packet sizes are variably sized between 46 and 1500 bytes (Din, 2009). As mentioned earlier, various overheads are added due to additional protocols employed for VPN tunnelling, thus, the space left for data shrinks so as not to break the 1500 byte MTU. Packet sizes can affect how quickly a packet is transmitted (Gilfeather, 2001), and relating back to encryption, also the time needed to encrypt the packet. As different protocols are going to be tested, it is inevitable that there will be further additional overheads for each protocol. It will be these additional overheads that will be investigated and compared for each VPN tunnelling method.

Latency is the time it takes a packet to travel from source to destination and is typically measured in milliseconds (ms). High latency delays can affect throughput performance as it takes longer for data to be transmitted (Kuang, 2010). A basic example of this is the noticeable delay after entering a web address, and the actual display of the page. As this investigation is based around remote access of a small business, it is conceivable that small organisations would have a relatively slow Internet connection as detailed earlier. Teleworkers would establish remote connections over the Internet, and this delay would need to be simulated in the experiment, which mentioned earlier, could have effects on the throughput of the VPN tunnel.

Chapter 3

Methods

This section will detail the specifics of the primary research approach used in this project. It will present a justification for this approach, as well as a detailed analysis of how the project will be carried out. Also discussed are the remaining activities required to complete this project.

3.1 Primary Research Method

To conduct the experimental project type it is possible to either undertake the experimentations using simulations or in a lab environment. For this project it is proposed that a lab-based experiment be utilised. There are many reasons why this environment should be used, the primary being realistic and accurate result sets.

By running experiments on physical hardware the mimicking of authentic small business scenarios, as discussed in Section 2.1.1, can be more accurately created. It is even possible for a router to be converted to a traffic generating device simulating typical traffic patterns of a small business detailed in Section 2.1.2. This can be recreated in a simulation environment; however, simulations can lack the nuances and quirks that occasionally occur in physical hardware. As (Flood, 1998) states, "*simulation results are only as good as the model and as such are still only estimates / projected outcomes*". By creating physical and life-life network scenarios, it is possible to adequately and truthfully reflect authentic data result sets, allowing for the best possible opportunity to draw well-founded and representative conclusions.

Another reason for pursuing an experimental based approach is the ability for others to advance or carry out further experiments with the system as described by (Truchan, 1993). A well-documented system can be easily recreated and additional experiments be conducted, allowing potential stakeholders to adapt the system to their specific needs (e.g. the ability of SSL VPNs in passing routing information between remote sites).

The alternative of using simulations to run the experimentations requires detailed knowledge of OPNET. OPNET is a comprehensive commercial simulation tool design to realistically mimic many different networking scenarios. Due to the individual undertaking the project having a limited knowledge of OPNET, but relatively easy access and comprehension of the configuration of physical hardware, it would be preferable to undertake this project using the lab-based

approach.

3.2 Intended Experiment

This section details the design of the devised small business network architecture, as well as the configuration and intended implementation of the experiment.

3.2.1 Architecture and Configuration

Based on the literature review conducted in Section 2.1.1, a typical small business architecture was found to contain a single routing and switching device requiring minimal configuration. Using equipment available to undertake this project, this design will be replicated using Cisco equipment and dual-core machines running Windows XP and Windows Server. As the author created Figure 3.1 denotes, a simple one router and one switch design, with multiple hosts connected to the switch, will be utilised to recreate the typical small office network layout. The servers will both be running Windows Server 2008 and configured with the latest versions of each VPN tunnelling method (PPTP server is pre-installed on Windows Server 2008 and OpenVPN server is currently at version 2.1.4 at time of writing).

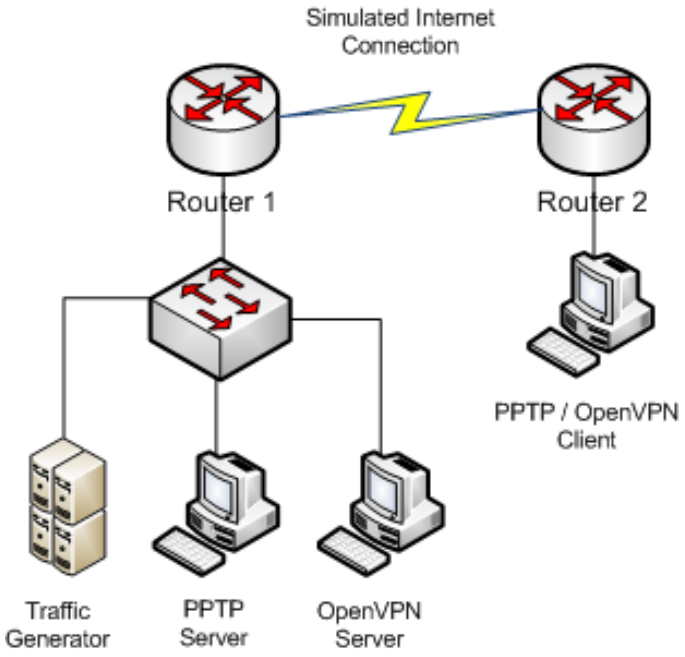


Figure 3.1: Experiment Diagram

Section 2.2.1 describes how VPNs are being utilised for remote connections into internal company networks. To simulate this, 2 Cisco 2811 routers will be used to create a *virtual* internet, complete with bandwidth and delay configurations. The configurations for Router 1 and Router 2 and can be found in Appendices A.1 and A.2 respectively. For the client side of the VPN configuration, a computer running Windows XP will be deployed with the latest client versions of PPTP (pre-installed on all versions of Windows Operating Systems) and OpenVPN (version 2.1.4 at time of writing). The OpenVPN configs required for both client and server can be found in Appendices A.8 and A.9. All hosts (i.e. clients and servers) will be connected to

their respective switch and router using FastEthernet 100Mbps links. The Cisco Catalyst 2960 switch will have no configuration modifications from its default settings which can be found in Appendix A.4.

The traffic generator in Figure 3.1 will be a Cisco 2811 router running a specialised Cisco Operating System complete with traffic generating abilities. This router has the ability to generate traffic to multiple destinations and includes several key features: burst traffic rates, multiple protocols, multiple port destinations, and variable packets sizing. The traffic patterns found during the literature review in Section 2.1.2 can all be simulated with this single device. The configuration script for this device can be seen in Appendix A.3.

3.2.2 Implementation

To test the performance of PPTP, it is intended to place various types of test data on the PPTP server, which will then be transferred to the client through the use of 3 different protocols: HTTP, HTTPS and SSH. HTTPS and SSH will be utilised to test the performance of *double-encryption*, as HTTPS and SSH will be already encrypted before the VPN tunnel encryption.

Each protocol will have their additional overheads, which discussed in Section 2.2.3, may result in different throughput values.

The test data will be comprised of various file types including generic binary files, pre-compressed zipped files, MPEG (Moving Picture Experts Group) videos, and PNG (Portable Network Graphics) images. Each file type will consist of multiple files totalling 100MB to provide an equal test base.

The time taken to transfer each individual file and the throughput will be recorded by the application used to transfer the file. For HTTP and HTTPS, the open-source tool *wget* will be employed to accurately track the time and average speed of each download. SSH file transfers will be conducted with the popular open-source terminal emulator $PuTTY$ due to its display of verbose statistics on throughput and transfer times.

The process for testing the performance of OpenVPN (SSL VPN) is almost identical to the PPTP experiment. As described in Section 2.2.2.2, OpenVPN has the ability to utilise different ciphers, each with different block sizes and algorithms, as well as on-the-fly compression of packets. The SSL VPN experiment will be run 6 times (compared to 1 run for PPTP) as the SSL VPN will be utilising 3 of the most commonly implemented ciphers, and variable compression states. Each *run* will consist of a transfer repeated 5 times to gain an average value for each performance metric. The following list defines the cipher and configuration settings that will be used for the 6 SSL VPN experiment runs:

- BF-CBC cipher (Blowfish 128 bit – OpenVPN default), no compression
- BF-CBC cipher (Blowfish 128 bit – OpenVPN default), with compression
- AES-128-CBC (AES 128 bit – moderately secure), no compression
- AES-128-CBC (AES 128 bit – moderately secure), with compression
- AES-256-CBC (AES 256 bit – most secure available), no compression

- AES-256-CBC (AES 256 bit – most secure available), with compression

BF-CBC (Blowfish) is selected as it is the default cipher enabled on OpenVPN connections. AES (with different key sizes) was chosen as it is known as one of the most secure ciphers available (as discussed in section 2.2.2.2). Two versions of this cipher are going to be tested to distinguish the performance differences between the *moderately* secure 128-bit key and the most secure 256-bit key versions. It should be noted that both version of AES are inherently secure and have no known *breakable* weaknesses.

The same test files are placed on the OpenVPN server and the client machine is swapped from PPTP to OpenVPN. Then each test file is transferred to the client through the SSL VPN tunnel for each protocol defined earlier.

To monitor the header sizes, and thus the overheads involved for each tunnelling method, the open source packet analyser *Wireshark* will be utilised. *Wireshark* can capture bits as they cross the wire and interpret specific details such as frame and packet headers, as well as the data being transmitted, and displays all the results in an intuitive graphical user interface – ideal for measuring specific header sizes quickly and easily.

Packet loss and latency will be constantly monitored using another open-source tool *WinMTR*. *WinMTR* is a network diagnostic tool that is able monitor the lowest, highest and average latency times, as well as the path (hops) to the specified destination. These results can then be saved in a number of formats including HTML and XML providing portability and readability.

After all 7 iterations of the experiments have concluded the results will be co-ordinated into a clear a concise tabular result set ready for evaluation.

As (Kuang, 2010) states that as the distance a packet travels directly correlates to the decrease of its throughput, the experiment will be repeated using a higher latency. However, the ability to artificially inflate the latency value is compromised due to affect this has on link speed. To increase the latency on the link the clock speed between the 2 routers has to be decreased, and as this occurs, it has the detrimental effect of decreasing the link speed. Therefore, to achieve the desired latency, the link speed will have to be decreased. To achieve a noticeable delay, the link bandwidth will be decrease to 2Mbps, resulting in an expected latency of around 100ms. The configurations used to achieve this on Router 1 and Router 2 can be found in Appendices A.5 and A.6 respectively. The amended traffic generation script used to reflect simulated traffic in the 2Mbps network can be found in Appendix A.7.

Chapter 4

Results

In this section the results gathered during the experiment, outlined in section 3.2, will be displayed and critically analysed based on the previously chosen performance metrics. Hypotheses identified in section 1.2.6 will also be confirmed or disproved based on the findings.

4.1 PPTP - 8Mbps (1024 KB/s)

4.1.1 HTTP

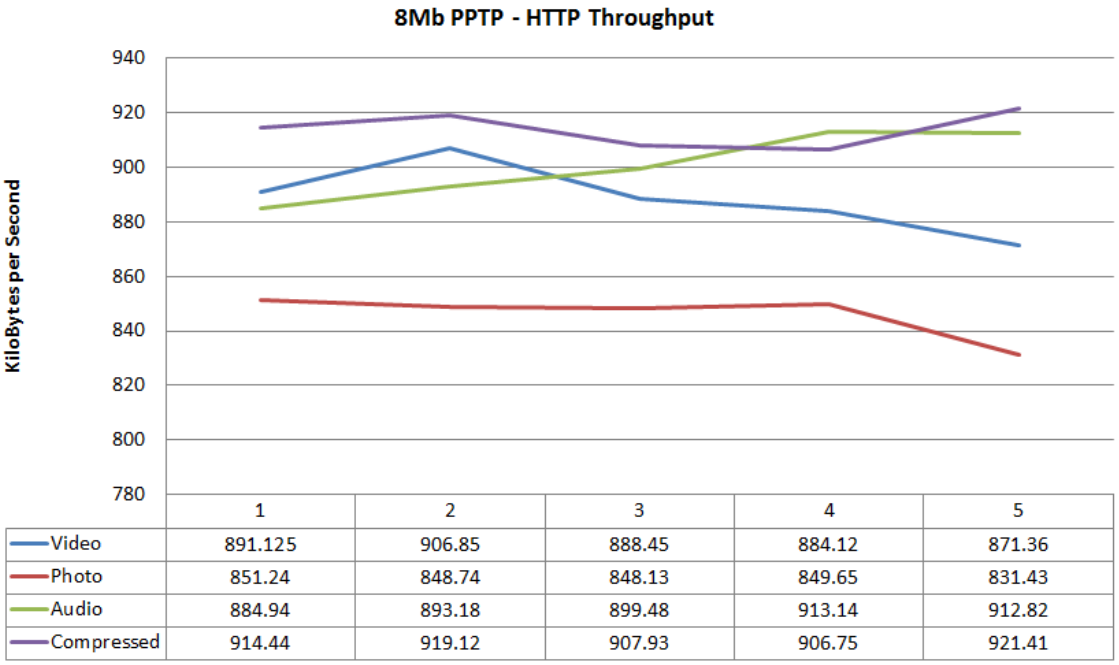
4.1.1.1 Throughput

Based on previous findings on VPN throughput performance conducted by (Khanvilkar & Khokhar, 2004), it was stated that the throughput never exceeds 50% of the original link bandwidth value. However, as expected, the throughput values for all file types easily exceed the 50% value recorded by (Khanvilkar & Khokhar, 2004).

As Figure 4.1 clearly shows, the photo test file has the lowest performance, with the compressed file achieving the best performance. For each test file, the difference between the 5 transfer iterations can be considered nominal (less than 5% between the maximum and minimum values during the video transfers), which indicates a stable and consistent network.

Average throughput and utilisation figures in Table 4.1 indicate very high performance characteristics with a value of greater than 80% link utilisation across all test files, including 89% for the compressed file. This value was anticipated to exceed 90% across many test files; however, it is possible that the simulated traffic being generated is causing throughput and utilisation values to decrease slightly.

As would be expected, the compressed file achieves the highest throughput value as it is able to transfer more bits in fewer packets due to its compressed nature. This is also true for audio and video files which have similar throughput and utilisation values due to their implementation of compression codecs.



	1	2	3	4	5
Video	891.125	906.85	888.45	884.12	871.36
Photo	851.24	848.74	848.13	849.65	831.43
Audio	884.94	893.18	899.48	913.14	912.82
Compressed	914.44	919.12	907.93	906.75	921.41

Figure 4.1: 8Mb PPTP HTTP Results

From these early results it is conceivable to conclude the pattern of file performance throughout the rest of the experiment. It is anticipated that compressed, video, audio, and photo will have throughput values of highest to lowest respectively across all further test runs.

File Type	**Avg. Throughput (KB/s)**	**Avg. Utilisation (%)**
Video	888.38	86.76
Photo	845.84	82.60
Audio	900.71	87.96
Compressed	913.93	89.25

Table 4.1: 8Mb PPTP HTTP Average Results

4.1.1.2 Latency and Jitter

As detailed in section 3.2, the latency and jitter is measured by the open-source program *WinMTR*. Table 4.2 displays the miniscule average latency differences and resulting jitter values recorded during the PPTP HTTP experiment run. The jitter values are so small they can be considered negligible and have no detrimental effect on the performance of each transfer.

Metric	**Run 1**	**Run 2**	**Run 3**	**Run 4**	**Run 5**
Latency	41ms	42ms	41ms	41ms	41ms
Jitter	0ms	1ms	0ms	0ms	0ms

Table 4.2: 8Mb PPTP HTTP Latency and Jitter Results

The latency values recorded in Table 4.2 are, again, indicative of a stable and consistent network. As the jitter values are so little, it can be concluded that the simulated traffic generated has no direct or negative affect of the VPN performance.

4.1.1.3 Packet Overhead

The packet structure was captured by another open-source program named *Wireshark* and appropriate results can be seen in Table 4.3.

As discussed in section 2.2.2.1, the minimum overhead incurred in PPTP is 24 bytes (20 bytes for IP header and 4 bytes for GRE); however, the GRE header was found to be 12 bytes in length – a 300% increase on its expected value. This was due to the preconfigured default PPTP options which include the use of checksums and sequence numbers. This will also increase the header to payload ratio of the overall packet, thus, each packet will contain less payload data due to the increase in header size.

Attribute	**Size (bytes)**
IP Header	20
GRE Header	12
Avg. Packet Size	697.65

Table 4.3: 8Mb PPTP HTTP Packet Overhead Results

The average packet size calculated by *Wireshark* during the HTTP tests returned a value of 697.65 bytes, revealing that the transmission of the test files were not efficient. This results in over 50% wastage (1500 bytes maximum) in packet space due to the dynamic sizing nature of IP packets – a seemingly inefficient transfer method. This results in an average header to payload ratio of 4.6%, instead of the expected 1.6% value under the most efficient scenario possible.

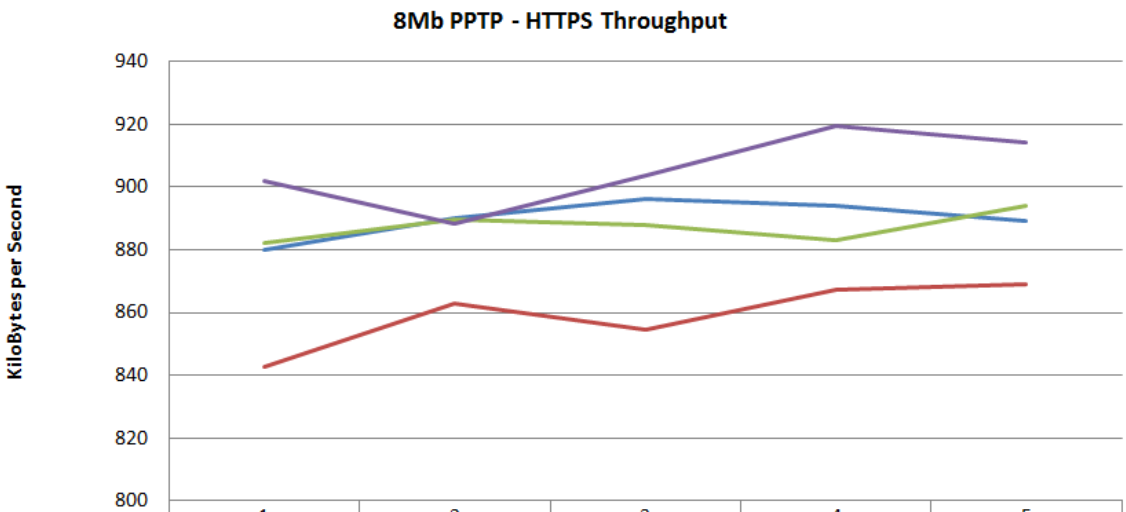
Whilst the 3% difference seems negligible, on slow links – such as those only available to small businesses – this number can translate in to additional kilobytes per second that could be used for business critical traffic. However, these figures denote the inherent problems in IP transmission packet sizing, not in PPTP. The PPTP overhead when compared to IP dynamic packet sizing could be considered negligible on large packets; nevertheless, as for any protocol in use, the smaller the packet size, the higher the overhead to payload ratio.

4.1.2 HTTPS

4.1.2.1 Throughput

As the difference between HTTP and HTTPS involves encryption, it is assumed that the results of the HTTPS transfers would decrease noticeably due to the processing delay of the encryption and decryption of all packets; however, the results in Figure 4.2 display remarkably similar, if not better, results in terms of throughput when compared to HTTP.

Figure 4.2 shows the compressed file achieving the fastest throughput, followed by video, audio, and then photo – very similar to HTTP in terms of overall position as well as throughput. Table 4.4 displays the average throughput and utilisation recorded, and shows almost all utilisation values within a 1% margin compared to the HTTP findings. It can be concluded from these results that double encryption does not affect throughput at all, apparently it increases performance! However, with a 1% margin, the resulting difference between HTTP and HTTPS can be considered negligible as they perform almost exactly the same.



	1	2	3	4	5
Video	879.93	889.95	896.12	894.26	889.11
Photo	842.62	862.87	854.56	867.49	869.24
Audio	882.12	889.64	887.84	883.22	894.16
Compressed	902.17	888.54	903.78	919.62	914.27

Figure 4.2: 8Mb PPTP HTTPS Results

File Type	**Avg. Throughput (KB/s)**	**Avg. Utilisation (%)**
Video	889.87	86.90
Photo	859.36	83.92
Audio	887.40	86.66
Compressed	905.68	88.44

Table 4.4: 8Mb PPTP HTTPS Average Results

Again, the findings of (Khanvilkar & Khokhar, 2004) do not hold true as all transfers are operating above 50% utilisation of the 8Mbps link. One possible reason that would explain the results would be that computing power has increased dramatically since 2004. Perhaps (Khanvilkar & Khokhar, 2004) were in fact limited not by VPN architecture, but by CPU power due to implementing large-block ciphers that required a large amount of resources to continually encrypt and decrypt all packets.

4.1.2.2 Latency and Jitter

As expected, based on the HTTP findings, Table 4.5 shows practically the same results as seen before. With jitter values non-existent or so low, they will have no effect, and it can be concluded that the simulated traffic generated has no direct or negative affect of the VPN performance.

Metric	**Run 1**	**Run 2**	**Run 3**	**Run 4**	**Run 5**
Latency	41ms	41ms	41ms	42ms	41ms
Jitter	0ms	0ms	0ms	1ms	0ms

Table 4.5: 8Mb PPTP HTTPS Latency and Jitter Results

4.1.2.3 Packet Overhead

Due to the HTTPS packet being encapsulated inside the PPP payload, the PPTP overhead will remain consistent with the HTTP findings. Any overheads incurred due to using HTTPS will not only be encrypted, but encapsulated inside the PPP payload leaving their contents obfuscated. However, as with HTTP latency and jitter results, the HTTPS results remain almost consistent as can be seen from Table 4.6.

Attribute	**Size (bytes)**
IP Header	20
GRE Header	12
Avg. Packet Size	683.47

Table 4.6: 8Mb PPTP HTTPS Packet Overhead Results

The one value that has changed is the average packet size. With a slightly decreased value, it could be speculated that HTTPS overheads (not visible due to encryption) slightly affect the packet size by adding additional overheads to the PPP payload. As (Morgan & Lovering, 2008) discusses, the symmetrical encryption that occurs in HTTPS is negligible in modern computing as computing power increases, thus, most overheads can be attributed to lengthy SSL handshakes. However, in the transfer of a single file, the initial handshake only occurs once during the creation of the SSL (HTTPS) connection; therefore, SSL handshakes are most likely not a factor in the performance of the PPTP VPN connection.

4.1.3 SSH

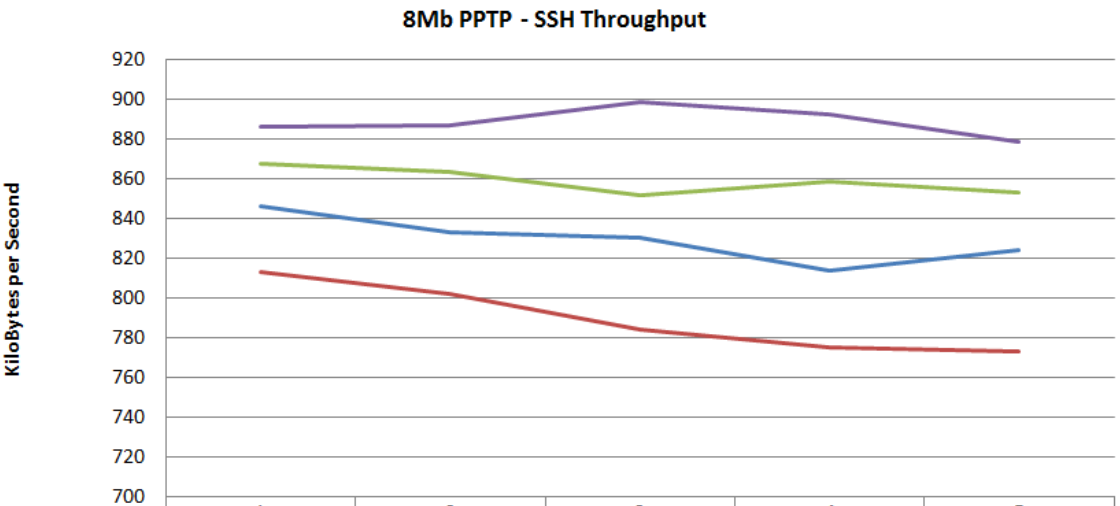
4.1.3.1 Throughput

SSH also involves the use of double encryption similar to HTTPS. Therefore, it was expected that the results would be similar to both HTTP and HTTPS finding. As Figure 4.3 displays, this assumption was partially correct as throughput and utilisation values are again fairly similar to earlier results; however, are slightly decreased.

Figure 4.3 shows the compressed file achieving the fastest throughput, followed by audio, video, and then photo – representative of earlier findings with audio and video achieving similar values throughout all tests conducted. Table 4.7 displays the average throughput and utilisation recorded, and shows more varied results, but still within a reasonable margin of 5% of HTTP and HTTP results. Unlike HTTPS, it could be speculated that double encryption performance by SSH does indeed affect throughput. However, just like HTTPS, a small margin such as 5% could be considered within the margin of error. Nevertheless, SSH does have a noticeable throughput difference that could be better utilised in other protocols.

File Type	**Avg. Throughput (KB/s)**	**Avg. Utilisation (%)**
Video	829.64	81.02
Photo	789.54	77.11
Audio	859.14	83.90
Compressed	888.76	86.79

Table 4.7: 8Mb PPTP SSH Average Results



	1	2	3	4	5
Video	846.25	833.56	830.39	813.86	824.13
Photo	813.29	802.05	784.15	775.25	772.96
Audio	867.76	863.76	852.17	858.78	853.22
Compressed	886.64	887.18	899.12	892.34	878.52

Figure 4.3: 8Mb PPTP SSH Results

Photo is once again the worst performing transfer, only utilising 77% of link bandwidth, most likely due to its relatively weak use or non-existence of pre-compression. Overall, throughput speeds are lower than both HTTP and HTTPS indicating a more intensive or inefficient encryption algorithm is being used. The results generated are consistent with initial expectations that double encryption would affect VPN performance.

4.1.3.2 Latency and Jitter

As with earlier findings, Table 4.8 shows almost the same results as seen before. With jitter values non-existent or so low, they will have no effect, and it can be concluded that the simulated traffic generated has no direct or negative affect of the VPN performance.

Metric	**Run 1**	**Run 2**	**Run 3**	**Run 4**	**Run 5**
Latency	41ms	41ms	41ms	41ms	42ms
Jitter	0ms	0ms	0ms	0ms	1ms

Table 4.8: 8Mb PPTP SSH Latency and Jitter Results

4.1.3.3 Packet Overhead

Similar to HTTPS, the SSH overheads are encapsulated inside the PPP payload, therefore, the PPTP overhead will remain consistent with HTTP and HTTPS findings. Table 4.9 shows almost no change from earlier results, other than the lower average packet size of 621.86 bytes.

The lower average packet size could lead to the speculation that SSH overheads (again, not visible due to encryption) affect the overall VPN performance as less data is being transferred per packet than with HTTP or HTTPS. It is likely (although not known) that the encryption

Attribute	Size (bytes)
IP Header	20
GRE Header	12
Avg. Packet Size	621.86

Table 4.9: 8Mb PPTP SSH Packet Overhead Results

method of SSH is less efficient than HTTPS, resulting in lower throughput and packet sizes, thus, affecting VPN performance and efficiency.

4.1.4 Conclusions

Overall, the results confirm that PPTP is a very efficient and high-performing VPN method with utilisation values averaging around 85%. Table 4.10 displays the average throughput and utilisation for each file type transferred, and shows that compressed files achieve very high performance characteristics above all other file types.

Protocol	Throughput (KB/s)	Utilisation (%)
Video	869.30	84.89
Photo	831.58	81.21
Audio	882.42	86.17
Compressed	902.79	88.16

Table 4.10: 8Mb PPTP Per File Type Summary Results

The photo file type was the worst performing transfer, most likely due to its lack of or inefficient use of compression algorithms. Video and audio files achieved fairly similar results consistently throughout all PPTP tests, with audio just out-performing video overall. The obvious winner is the compressed file type due to its advanced pre-compression techniques, which results in more data being transferred in a shorter amount of time.

Jitter and latency were consistent throughout all PPTP tests, with a maximum of 1ms of jitter and 42ms of latency. Without comparison against a higher latency link, it is unclear whether latency does indeed affect throughput performance as (Kuang, 2010) states.

As was proven various times throughout the PPTP results, (Khanvilkar & Khokhar, 2004) findings that different VPN methods do not exceed 50% utilisation of the link speed were found to be incorrect, or at least, not able to be replicated. This may be due to their test being conducted in 2004, when CPU power (as dictated by Moore's Law), was significantly less powerful than today in 2011.

Overall, PPTP, despite being one of the oldest, is an extremely reliable and efficient VPN tunnelling method still in large-scale use today. With an average utilisation of 85% across a range of file types and protocols, it can be concluded that a 15% drop in (theoretical) performance (i.e. the wire-speed, assuming no overheads), is a performance level sufficient enough to warrant its use by organisations requiring encryption to safeguard secret and confidential information.

However, as stated in section 2.2.2.1, PPTP is not cryptographically secure and is vulnerable to various methods of attack. Therefore, it is not recommended as a primary method of VPN tunnelling; nevertheless, it is a viable alternative for remote access when encryption is not

essential.

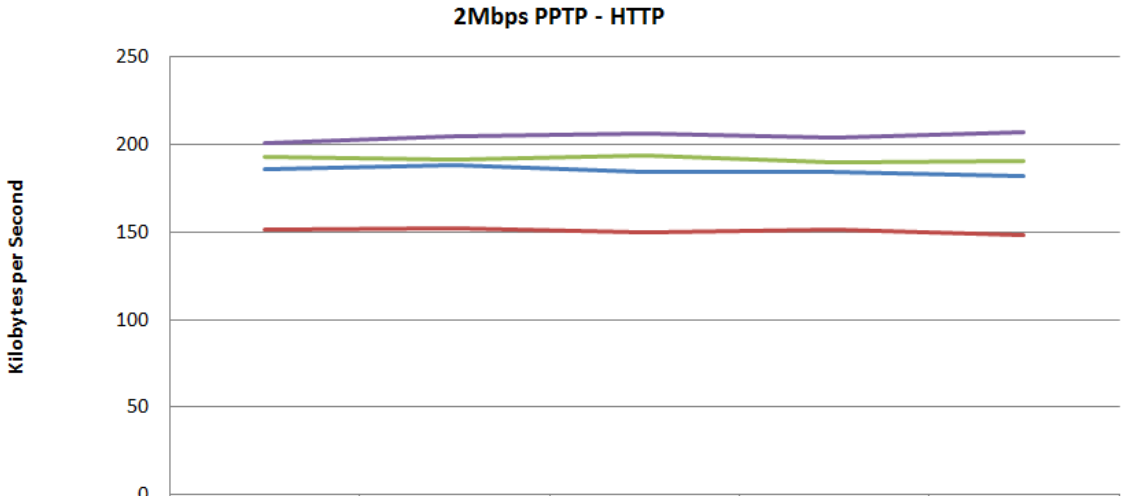
4.2 PPTP - 2Mbps (256 KB/s)

To determine if latency does affect throughput as (Kuang, 2010) states, it is necessary to increase the latency on the link between Router 1 and Router 2. However, the configuration required to introduce this latency also directly interferes with the bandwidth link speed. To introduce a noticeable latency of over 100ms, the bandwidth of the link if required to be set to 2Mbps. Throughput results gathered during the 2Mbps PPTP test can still be compared to the 8Mbps findings by using utilisation values to compare VPN performance over a simulated long distance (high latency).

4.2.1 HTTP

4.2.1.1 Throughput

According to (Kuang, 2010), latency should have a directly correlation to the decrease of throughput on a network, where high latency is indicative of a long-distance connection. The results gathered and displayed in Figure 4.4 support this conclusion. With a maximum theoretical throughput of 256 KB/s, the highest average value achieved was 204.69 KB/s during the compressed file transfer as shown in Table 4.11.



	1	2	3	4	5
Video	185.86	188.59	184.66	184.70	181.74
Photo	151.45	152.62	150.06	151.64	148.36
Audio	193.41	191.10	193.69	190.27	190.65
Compressed	201.14	204.84	206.17	204.01	207.27

Figure 4.4: 2Mb PPTP HTTP Results

As with previous results, the position of each throughput value has been consistent, with the compressed file transferring the fastest, followed closely by both audio and video. The photo transfer trails in last with a noticeable throughput gap emerging from the rest of the transfers.

Table 4.11 also shows that the average utilisation values for all transfers have dropped fairly drastically, over 10% from all respective 8Mbps PPTP results.

File Type	Avg. Throughput (KB/s)	Avg. Utilisation (%)
Video	185.11	72.31
Photo	150.83	58.92
Audio	191.82	74.93
Compressed	204.686	79.96

Table 4.11: 2Mb PPTP HTTP Average Results

The most substantial drop is performance occurs during the photo transfer, where the utilisation has dropped almost 25%. All other file transfers suffered from a 10-15% decrease in link utilisation. These results would tend to agree with the theory by (Kuang, 2010) that latency affects throughput speeds. However, further analysis of the other protocols is required before a final conclusion is made.

4.2.1.2 Latency and Jitter

Similar to the 8Mbps PPTP results, the latency and jitter values remained constant as can be seen in Table 4.12, suggesting that the simulated traffic generated has no direct or negative affect of the VPN performance, as well as confirming the networks stability.

Metric	Run 1	Run 2	Run 3	Run 4	Run 5
Latency	115ms	115ms	115ms	115ms	115ms
Jitter	0ms	0ms	0ms	0ms	0ms

Table 4.12: 2Mb PPTP HTTP Latency and Jitter Results

The two differences that occurred compared to results previously generated are the increased latency recorded at 115ms, and the jitter which is now non-existent and reads a value of zero. At this stage it does appear that the increased latency affects the throughput as (Kuang, 2010) stated; however, further testing is required to confirm this conclusion.

4.2.1.3 Packet Overhead

Since PPTP is still in use, the contents of the header inside the PPP payload are unreadable, thus, the only known overheads are listed in Table 4.13. The overheads and their respective sizes still remain the same in line with all 8Mbps PPTP results.

Attribute	Size (bytes)
IP Header	20
GRE Header	12
Avg. Packet Size	711.38

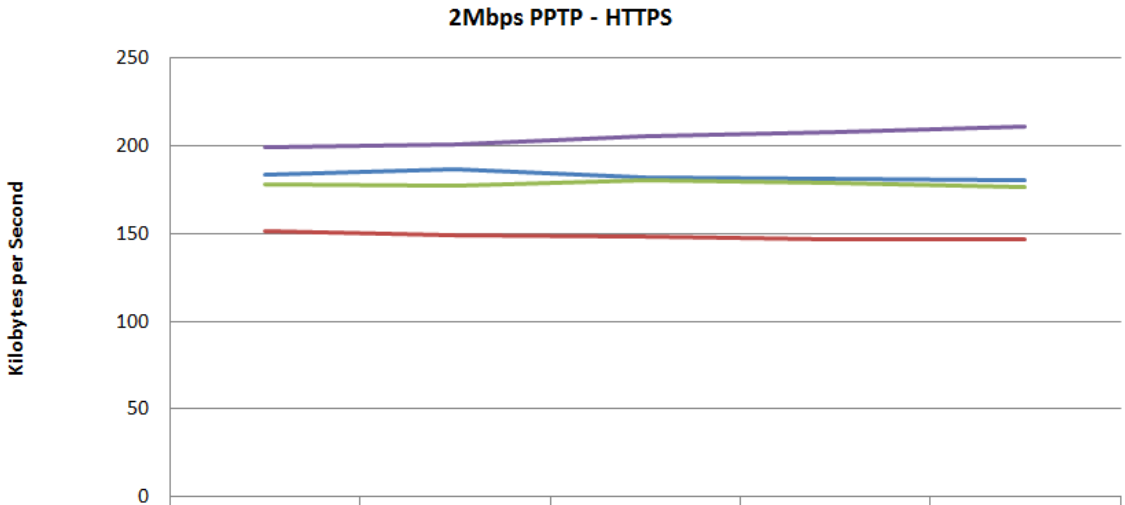
Table 4.13: 2Mb PPTP HTTP Packet Overhead Results

The only difference is the average packet size which was identified by *Wireshark* as 711.38 bytes. This is a slight increase compared to 8Mbps PPTP HTTP result of 697.65 bytes. The difference in values is around 1%, a miniscule amount which can be considered negligible.

4.2.2 HTTPS

4.2.2.1 Throughput

Based on the results analysed from the 8Mbps PPTP HTTP and HTTPS tests, it would be reasonable to speculate that the 2Mbps HTTP and HTTPS results would also be extremely similar. The numbers gathered and displayed in Figure 4.5 and Table 4.14 support this reasoning and show almost identical values within a 1-2% margin of the 2Mbps HTTP results.



	1	2	3	4	5
Video	183.93	186.57	182.01	181.00	180.64
Photo	151.78	149.02	148.39	146.48	147.01
Audio	178.45	177.49	180.47	179.26	176.85
Compressed	199.12	200.98	205.51	207.78	211.34

Figure 4.5: 2Mb PPTP HTTPS Results

Only the audio transfer failed to stay within the 1-2% margin compared to HTTP results; however, the 5% margin and previous analysis of 8Mbps results allows the conclusion that the value is only a slight discrepancy, and cannot be considered indicative of a definitive decline in audio transfer throughput on slower links.

File Type	**Avg. Throughput (KB/s)**	**Avg. Utilisation (%)**
Video	182.83	71.42
Photo	148.54	58.02
Audio	178.50	69.73
Compressed	204.95	80.06

Table 4.14: 2Mb PPTP HTTPS Average Results

Similar to the 2Mbps PPTP HTTP results (see Figure 4.4), the results show relatively no change in the average throughput rate, as indicated by the almost straight lines for every file type transferred – also indicative of a stable and consistent network.

4.2.2.2 Latency and Jitter

The latency and jitter values remained constant as can be seen in Table 4.15, suggesting that the simulated traffic generated has no direct or negative affect of the VPN performance, as well as confirming the networks stability. These results are also supportive of previous findings during the 2Mbps PPTP HTTP test.

Metric	**Run 1**	**Run 2**	**Run 3**	**Run 4**	**Run 5**
Latency	115ms	115ms	115ms	115ms	115ms
Jitter	0ms	0ms	0ms	0ms	0ms

Table 4.15: 2Mb PPTP HTTPS Latency and Jitter Results

There are no discernable differences in latency or jitter between the results gathered throughout the 2Mbps PPTP tests.

4.2.2.3 Packet Overhead

Since PPTP is still in use, the contents of the header inside the PPP payload are unreadable, thus, the only known overheads are listed in Table 4.16. The overheads and their respective sizes still remain the same in line with all 8Mbps and 2Mbps PPTP results.

Attribute	**Size (bytes)**
IP Header	20
GRE Header	12
Avg. Packet Size	704.72

Table 4.16: 2Mb PPTP HTTPS Packet Overhead Results

The only difference is the average packet size which is identified as 704.72 bytes. This is a slight decrease compared to the 2Mbps PPTP HTTP result of 711.38 bytes. The difference in values, once again, is around 1%, an amount so small it can be disregarded.

4.2.3 SSH

4.2.3.1 Throughput

The 8Mb PPTP SSH results indicated a noticeable decrease in performance compared to its relative HTTP and HTTPS results. It was speculated that this trend would continue on the high-latency 2Mb PPTP SSH test. Figure 4.6 displays fairly decreased throughput rates, with almost no deviation in average throughput rates between all 5 iterations.

As expected, Figure 4.6 shows the compressed file achieving the fastest throughput, followed by audio, video, and then photo – representative of earlier findings with audio and video achieving similar values throughout all tests conducted. Table 4.17 displays the average throughput and utilisation recorded, and shows significant decreases in throughput and utilisations compared to the 8Mbps PPTP SSH results. Thus, just like all other 2Mbps PPTP results, show a sharp

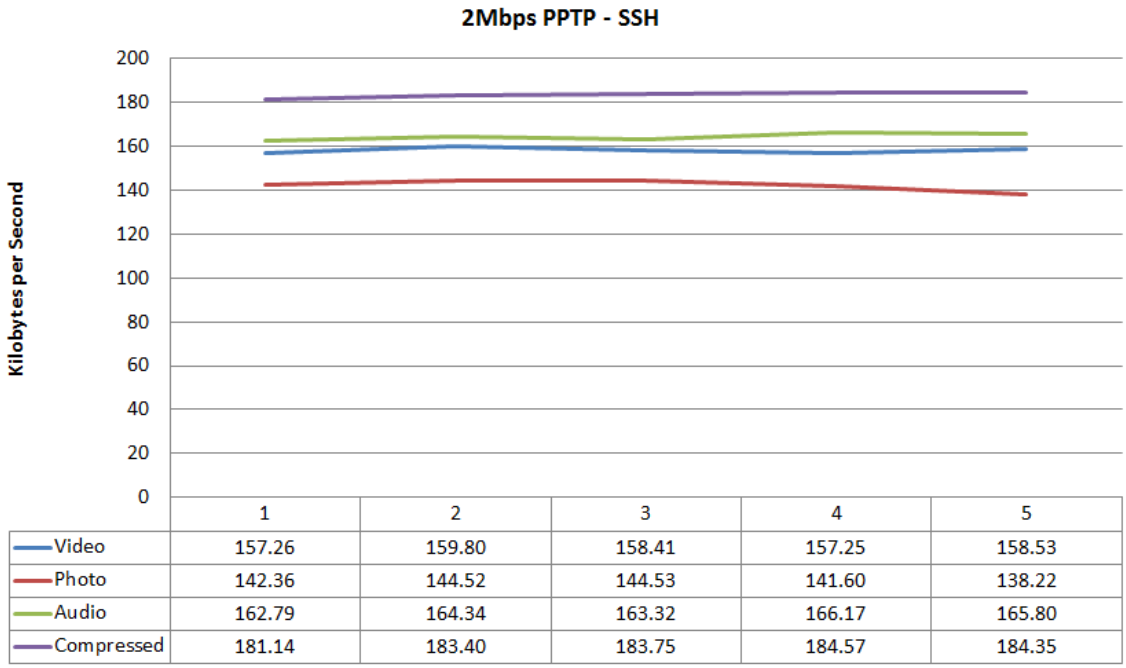


Figure 4.6: 2Mb PPTP SSH Results

File Type	Avg. Throughput (KB/s)	Avg. Utilisation (%)
Video	158.25	61.82
Photo	142.25	55.56
Audio	164.48	64.25
Compressed	183.44	71.66

Table 4.17: 2Mb PPTP SSH Average Results

decline in performance on low-speed (high-latency) links. Another reason for its relatively bad performance could be that double encryption is affecting throughput as initially theorised.

Photo is once again the worst performing transfer, only marginally sitting above the 50% barrier discussed by (Khanvilkar & Khokhar, 2004). Overall, throughput speeds are lower than both HTTP and HTTPS indicating a more intensive or inefficient encryption algorithm is being used. The results generated are consistent with initial expectations that double encryption would affect VPN performance.

4.2.3.2 Latency and Jitter

The latency and jitter values remained constant as can be seen in Table 4.18, suggesting that the simulated traffic generated has no direct or negative affect of the VPN performance, as well as confirming the networks stability. These results are also supportive of previous findings during the 2Mbps PPTP HTTP and HTTPS tests.

There are no discernable differences in latency or jitter between the results gathered throughout the 2Mbps PPTP tests.

Metric	Run 1	Run 2	Run 3	Run 4	Run 5
Latency	115ms	115ms	115ms	115ms	115ms
Jitter	0ms	0ms	0ms	0ms	0ms

Table 4.18: 2Mb PPTP SSH Latency and Jitter Results

4.2.3.3 Packet Overhead

The only known and visible overheads are listed in Table 4.19. The overheads and their respective sizes still remain the same in line with all 8Mbps and 2Mbps PPTP results.

Attribute	Size (bytes)
IP Header	20
GRE Header	12
Avg. Packet Size	655.16

Table 4.19: 2Mb PPTP SSH Packet Overhead Results

The only difference is the average packet size which is identified as 655.16 bytes. This is a slight decrease compared to the 2Mbps PPTP HTTPS result of 704.72 bytes. The difference in values is around 3-4%, an amount small enough it can be considered inconsequential.

4.2.4 Conclusions

Overall, the results confirm the statement by (Kuang, 2010) that latency directly affects throughput. Compared to the 8Mbps PPTP results, the 2Mbps results reveal PPTP is not as efficient or high-performing at slow speeds as it is only capable of averaging 68% utilisation – almost 20% lower. Table 4.20 displays the average throughput and utilisation for each file type transferred, and shows that compressed files achieve relatively high performance characteristics above all other file types.

Protocol	Throughput (KB/s)	Utilisation (%)
Video	174.92	68.33
Photo	149.17	58.27
Audio	180.12	70.36
Compressed	196.64	76.81

Table 4.20: 2Mb PPTP Per File Type Summary Results

Once again, the photo file type was the worst performing transfer, most likely due to its lack of or inefficient use of compression algorithms. Video and audio files achieved fairly similar results consistently throughout all PPTP tests, with audio just out-performing video overall. The obvious winner is the compressed file type due to its advanced pre-compression techniques, which results in more data being transferred in a shorter amount of time.

Jitter and latency were consistent throughout all 2Mb PPTP tests, with a maximum value of 115ms for latency and zero jitter.

As was proven various times throughout the PPTP results, (Khanvilkar & Khokhar, 2004) findings that different VPN methods do not exceed 50% utilisation of the link speed were

found to be incorrect, or at least, not able to be replicated. This may be due to their test being conducted in 2004, when CPU power (as dictated by Moore's Law), was significantly less powerful than today in 2011. However, the results for the 2Mb PPTP tests definitively displayed a noticeable decrease in throughput when transferring over a high-latency link – with the photo transfer close to the 50% value stated by (Khanvilkar & Khokhar, 2004).

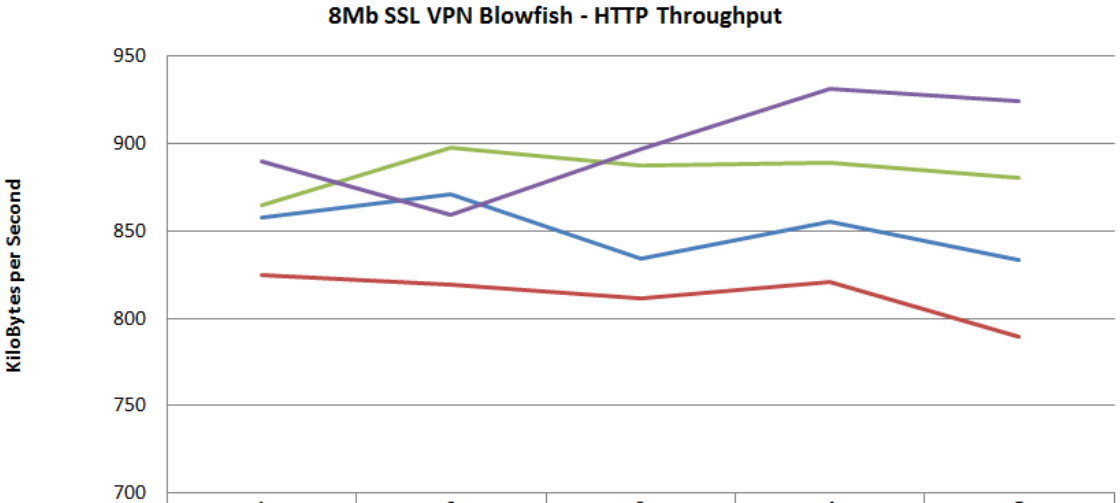
With an average utilisation of only 68% across a range of file types and protocols on the 2Mbps link, it can be concluded that the 32% drop in (theoretical) performance (i.e. the wire-speed, assuming no overheads), is a significant performance drop compared to higher-speed links. A 32% reduction of a 2Mbps link with a theoretical maximum throughput of 256 KB/s equates to 174 KB/s – a substantial drop of over 80 KB/s. This is the equivalent of 20 16Kbps Voice Over IP phone calls, throughput which could be better utilised providing business critical services. Therefore, it is not recommended that PPTP be used over slow-speed or high latency links unless security or performance is not a factor.

4.3 SSL VPN - 8Mb (1024 KB/s) - Blowfish

4.3.1 HTTP

4.3.1.1 Throughput

Blowfish is a 128-bit cipher employed as the default cipher used by OpenVPN. As Figure 4.7 clearly shows, the photo test file once again has the lowest performance, with the compressed file achieving the best performance. The values for each iteration are fairly consistent and are reasonably similar to previous results from the 8Mb PPTP HTTP test.



	1	2	3	4	5
Video	857.93	871.45	834.08	855.11	833.57
Photo	824.69	819.18	811.85	820.63	789.41
Audio	864.84	898.06	887.50	888.89	880.72
Compressed	889.54	859.00	896.74	931.61	924.41

Figure 4.7: 8Mb SSL VPN Blowfish HTTP Results

Average throughput and utilisation figures in Table 4.21 indicate very high performance char-

acteristics with a value of greater than 80% link utilisation across most test files, including 88% for the compressed file. Similar to 8Mb PPTP predictions, this value was anticipated to exceed 90% across many test files; however, it is possible that the simulated traffic being generated is causing throughput and utilisation values to decrease slightly.

As would be expected, the compressed file achieves the highest throughput value as it is able to transfer more bits in fewer packets due to its compressed nature. This is also true for audio and video files which have fairly similar throughput and utilisation values due to their implementation of compression codecs.

From these early results it is conceivable to conclude the pattern of file performance throughout the rest of the experiment. It is anticipated that compressed, video, audio, and photo will have throughput values of highest to lowest respectively across all further test runs.

File Type	**Avg. Throughput (KB/s)**	**Avg. Utilisation (%)**
Video	857.93	83.05
Photo	824.69	79.41
Audio	864.84	86.33
Compressed	889.54	87.92

Table 4.21: 8Mb SSL VPN Blowfish HTTP Average Results

4.3.1.2 Latency and Jitter

Similar to previous 8Mb PPTP tests, Table 4.22 shows almost the same results as seen before. With jitter values non-existent or so low, they will have no effect, and it can be concluded that the simulated traffic generated has no direct or negative affect of the VPN performance.

Metric	**Run 1**	**Run 2**	**Run 3**	**Run 4**	**Run 5**
Latency	41ms	41ms	41ms	41ms	41ms
Jitter	0ms	0ms	0ms	0ms	0ms

Table 4.22: 8Mb SSL VPN Blowfish HTTP Latency and Jitter Results

4.3.1.3 Packet Overhead

As discussed in section 2.2.2.2, the packet structure and overheads of OpenVPN are already known. These values were verified by the use of *Wireshark*, and the values collected can be viewed in Table 4.23.

Attribute	**Size (bytes)**
IP Header	20
UDP Header	8
Security Layer	41
Avg. Packet Size	391.85

Table 4.23: 8Mb SSL VPN Blowfish HTTP Packet Overhead Results

The security layer (known to be 41 bytes – see Section 2.2.2.2) is encapsulated inside the frame payload, thus, the size cannot be verified. However, for calculation purposes, this figure will be included in all calculations.

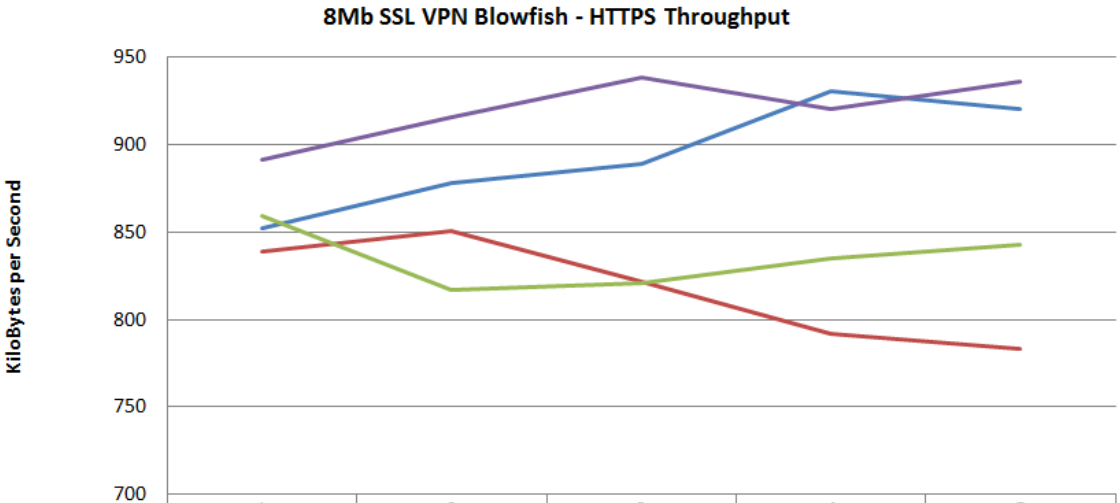
Visible in Table 4.23 is a much smaller average packet size compared to previous test results. Such small packets will result in ballooning header to payload ratios, further decreasing SSL VPN efficiency. A packet size of 391.85 bytes equates to 26.12% of every packet consisting of headers – a huge increase compared to the 4.6% value observed in the 8Mb PPTP HTTP results.

However, it initially seems that the increased overheads on every packet do not have a large effect on throughput speeds, as is evident by the maximum throughput of 889.54 KB/s compared to the 8Mb PPTP HTTP value of 913.93 KB/s.

4.3.2 HTTPS

4.3.2.1 Throughput

Similar to the 8Mb PPTP HTTPS test results, Figure 4.7 displays improvements on some file transfers compared to their relative HTTP results, indicating double encryption does not factor into VPN performance. As with most other results, the compressed files has the highest throughput rate followed by video, audio, and photo with the slowest performance. However, unlike previous results, there is a large gap between the video and audio throughputs – almost 60 KB/s.



	1	2	3	4	5
Video	852.52	878.38	889.06	930.54	920.57
Photo	838.58	850.85	821.54	792.12	783.67
Audio	859.63	816.89	820.95	834.66	842.98
Compressed	891.42	915.98	938.29	920.67	935.87

Figure 4.8: 8Mb SSL VPN Blowfish HTTPS Results

Average throughput and utilisation figures in Table 4.24 indicate very high performance characteristics with a value of greater than 80% link utilisation across most test files, including 90% for the compressed file.

It could be speculated that the results are following the same pattern as the 8Mb PPTP results as values between the two are marginal and all within the 15% margin theorised as part of the hypotheses in Section 1.2.6. It could also be argued that due to such similar performance

characteristics, the more advanced Blowfish cipher employed in OpenVPN has little to no effect on VPN performance – most likely due to the greater CPU power available now than (Khanvilkar & Khokhar, 2004) or (Hall, 2008) had access to at the time of their performance related experiments.

File Type	Avg. Throughput (KB/s)	Avg. Utilisation (%)
Video	894.21	87.33
Photo	817.35	79.82
Audio	835.02	81.55
Compressed	920.45	89.89

Table 4.24: 8Mb SSL VPN Blowfish HTTPS Average Results

4.3.2.2 Latency and Jitter

Table 4.25 shows results consistent with previous findings. With jitter values non-existent or so low, they will have no effect, it can be concluded that the simulated traffic generated has no direct or negative affect of the VPN performance.

Metric	Run 1	Run 2	Run 3	Run 4	Run 5
Latency	41ms	41ms	41ms	41ms	41ms
Jitter	0ms	0ms	0ms	0ms	0ms

Table 4.25: 8Mb SSL VPN Blowfish HTTPS Latency and Jitter Results

4.3.2.3 Packet Overhead

The header values have remained constant between 8Mb SSL VPN tests, as can be seen in Table 4.26.

Attribute	Size (bytes)
IP Header	20
UDP Header	8
Security Layer	41
Avg. Packet Size	384.20

Table 4.26: 8Mb SSL VPN Blowfish HTTPS Packet Overhead Results

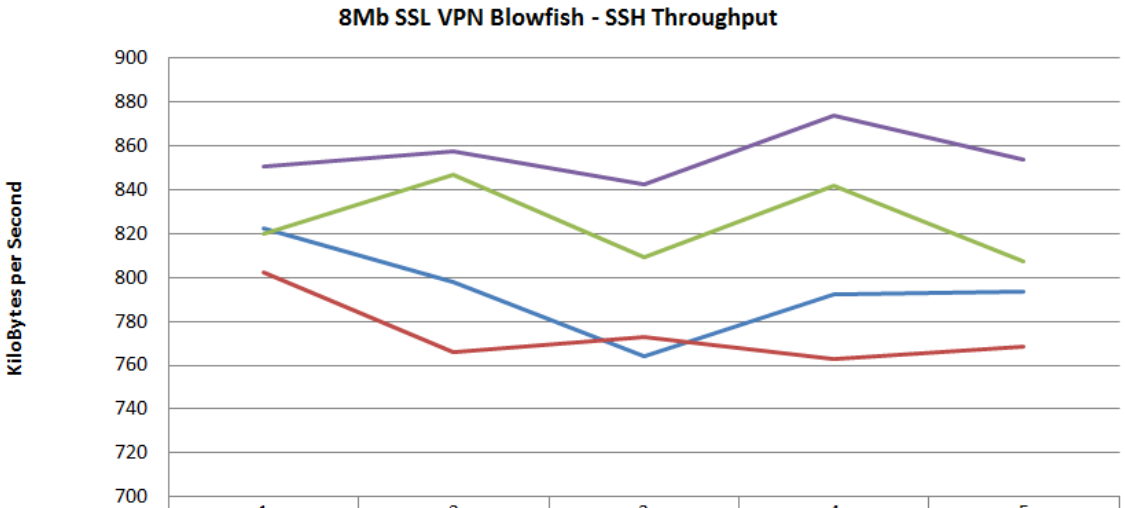
Table 4.26 shows a slight decrease in the average packet size – down to 384.20 bytes. This could be attributed to the double encryption and extra overhead that HTTPS incurs. However, this only equates to around a 1% decrease and is so miniscule, it can be disregarded.

Following on from initial speculations, it appears that packet size does not have a noticeable effect on VPN performance. Nevertheless, it is too early to form a conclusion as this theory must first be tested on various protocols to be substantiated.

4.3.3 SSH

4.3.3.1 Throughput

It is anticipated that the extra inefficiencies and encryption overheads identified during the 8Mb PPTP SSH tests will also affect the overall VPN performance of the SSL VPN Blowfish SSH results. This can be confirmed by the results gathered and visible in Figure 4.9. Average throughput and utilisation figures in Table 4.27 indicate relatively high performance characteristics within the 15% margin specified in Section 1.2.6.



	1	2	3	4	5
Video	822.36	797.69	763.82	792.62	793.39
Photo	802.47	765.69	772.67	763.13	768.60
Audio	819.97	847.05	809.30	841.86	807.53
Compressed	850.42	857.80	842.61	874.17	853.70

Figure 4.9: 8Mb SSL VPN Blowfish SSH Results

Table 4.27 displays values that complement the findings of the 8Mb PPTP SSH test, as all values fall within a 5% margin, which could be argued that it falls within a respectable margin of error. This would indicate that the performance difference of SSH over PPTP and SSL VPNs with the Blowfish cipher is nominal – as could be argued about the HTTP and HTTPS protocols too. However, it is too early to speculate on how higher-grade ciphers will fair when implementing the same protocols.

File Type	**Avg. Throughput (KB/s)**	**Avg. Utilisation (%)**
Video	793.98	77.54
Photo	774.51	75.64
Audio	825.14	80.58
Compressed	855.74	83.57

Table 4.27: 8Mb SSL VPN Blowfish SSH Average Results

4.3.3.2 Latency and Jitter

Table 4.28 shows results consistent with previous findings. With jitter values non-existent or so low, they will have no effect, it can be concluded that the simulated traffic generated has no direct or negative affect of the VPN performance.

Metric	Run 1	Run 2	Run 3	Run 4	Run 5
Latency	41ms	41ms	41ms	41ms	41ms
Jitter	0ms	0ms	0ms	0ms	0ms

Table 4.28: 8Mb SSL VPN Blowfish SSH Latency and Jitter Results

4.3.3.3 Packet Overhead

As previous discovered, the header values have remained constant between 8Mb SSL VPN tests, as can be seen in Table 4.29.

Attribute	Size (bytes)
IP Header	20
UDP Header	8
Security Layer	41
Avg. Packet Size	379.13

Table 4.29: 8Mb SSL VPN Blowfish SSH Packet Overhead Results

Table 4.29 shows another slight decrease in the average packet size – down to 379.13 bytes. This could be attributed to the double encryption and inefficiencies previously mentioned that SSH incurs. However, this only equates to around a 1% decrease and is so miniscule, it can be disregarded.

4.3.4 Conclusions

After concluding the 8Mb SSL VPN Blowfish tests, it is fairly clear that the performance difference between PPTP and SSL VPN with the Blowfish cipher is relatively insignificant. Tables 4.30 and 4.31 clearly display the similarities in performance between the two different VPN tunnelling methods.

Protocol	Throughput (KB/s)	Utilisation (%)
Video	837.49	81.79
Photo	825.14	80.58
Audio	852.36	83.24
Compressed	887.14	86.64

Table 4.30: 8Mb SSL VPN Blowfish Per File Type Summary Results

For each file type transferred, the utilisation between the two VPN tunnelling methods is within a 3-4% margin. This can be considered relatively minute and indicates that SSL VPNs with Blowfish (a secure cipher) has almost the same performance attributes as PPTP with its known weak encryption cipher. Compound this advantage with the fact that OpenVPN is open-source

Protocol	Throughput (KB/s)	Utilisation (%)
Video	869.30	84.89
Photo	831.58	81.21
Audio	882.42	86.17
Compressed	902.79	88.16

Table 4.31: 8Mb PPTP Per File Type Summary Results

and requires no licenses to operate, whereas PPTP typically requires a Windows Server with appropriate licensing to operate, it is perfect for small businesses that require modern, but cost appropriate IT technology.

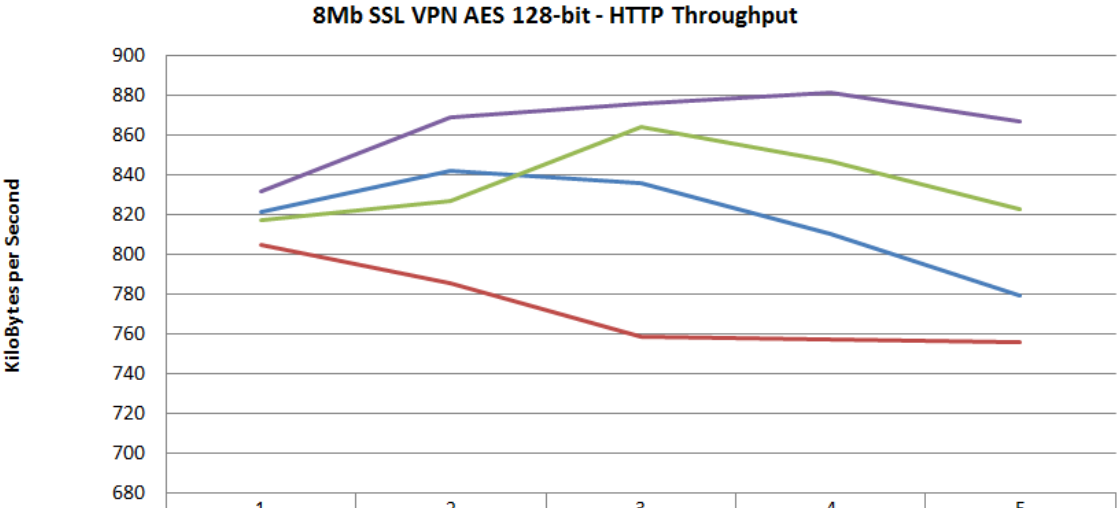
However, higher grade ciphers can be implemented within OpenVPN, resulting in a much more secure and robust VPN tunnelling, whilst – hopefully – retaining a reasonable performance margin against PPTP of 15%, as defined in the hypotheses in Section 1.2.6.

4.4 SSL VPN - 8Mb (1024 KB/s) - AES 128-bit

4.4.1 HTTP

4.4.1.1 Throughput

Next, AES 128-bit is being tested because whilst maintaining the same key size as Blowfish, AES is a much more advanced encryption cipher which requires additional computation (Lu & Tseng, 2002). It could be speculated from Figure 4.10 that this statement holds true as throughput values have decreased across all file transfers.



	1	2	3	4	5
Video	821.69	842.59	835.84	810.62	779.54
Photo	804.84	785.95	758.55	757.14	755.81
Audio	817.56	826.87	864.23	847.23	822.88
Compressed	832.01	869.35	876.43	881.64	867.19

Figure 4.10: 8Mb SSL VPN AES 128-bit HTTP Results

Table 4.32 shows the compressed file achieving the fastest throughput, followed by audio, video, and then photo – representative of earlier findings with audio and video achieving similar values throughout all tests conducted. However, average throughput and utilisation values have dropped by roughly 5% compared to Blowfish results – most likely due to the aforementioned additional computation required for the AES cipher.

File Type	**Avg. Throughput (KB/s)**	**Avg. Utilisation (%)**
Video	818.06	79.89
Photo	772.46	75.43
Audio	835.75	81.62
Compressed	865.32	84.50

Table 4.32: 8Mb SSL VPN AES 128-bit HTTP Average Results

It is clear that the AES 128-bit cipher is having a noticeable effect on performance. However, with only a 5% drop in performance against Blowfish results, AES 128-bit findings are still indicative that SSL VPNs will perform within a 15% margin of PPTP.

4.4.1.2 Latency and Jitter

Table 4.33 shows results consistent with previous findings. With jitter values non-existent or so low, they will have no effect, it can be concluded that the simulated traffic generated has no direct or negative affect of the VPN performance.

Metric	**Run 1**	**Run 2**	**Run 3**	**Run 4**	**Run 5**
Latency	41ms	41ms	41ms	41ms	41ms
Jitter	0ms	0ms	0ms	0ms	0ms

Table 4.33: 8Mb SSL VPN AES 128-bit HTTP Latency and Jitter Results

4.4.1.3 Packet Overhead

Header values have remained constant between all 8Mb SSL VPN tests so far, as can be seen in Table 4.34.

Attribute	**Size (bytes)**
IP Header	20
UDP Header	8
Security Layer	41
Avg. Packet Size	402.24

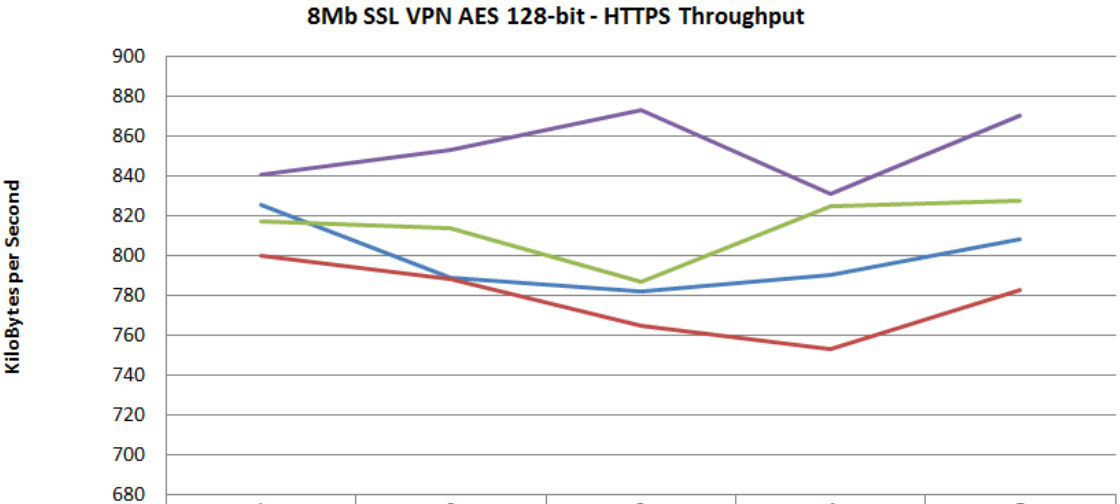
Table 4.34: 8Mb SSL VPN AES 128-bit HTTP Packet Overhead Results

Table 4.34 shows an average packet size similar to that of the earlier Blowfish results. This would be indicative that packet size does not have a direct correlation to throughput, as the average packet size has actually increased, but overall performance has dropped slightly due to the additional crypto computations.

4.4.2 HTTPS

4.4.2.1 Throughput

Following the pre-determined results from previous HTTPS results, it is speculated that the HTTPS throughput values would be similar (within a 2% margin) to its relative HTTP results. This is evident in 4.11 as it shows all transfer throughputs and utilisation values sitting within this margin.



	1	2	3	4	5
Video	825.85	789.33	782.23	790.79	808.82
Photo	800.12	788.16	765.30	753.61	782.69
Audio	817.22	813.79	787.00	824.85	827.63
Compressed	841.02	853.46	873.57	831.05	870.87

Figure 4.11: 8Mb SSL VPN AES 128-bit HTTPS Results

Once again, Table 4.35 indicates that the compressed file achieves the fastest throughput, followed by audio, video, and then photo. As expected, the HTTPS results are consistent with previous PPTP and SSL VPN Blowfish results, which show very little performance difference between the HTTP and HTTPS protocols. This could be attributed to a less-CPU intensive cipher being employed by HTTPS, or that CPU power is sufficient enough as not to add additional processing delay on encryption and decryption of packets.

File Type	**Avg. Throughput (KB/s)**	**Avg. Utilisation (%)**
Video	799.40	78.07
Photo	777.98	75.97
Audio	814.10	79.50
Compressed	853.99	83.40

Table 4.35: 8Mb SSL VPN AES 128-bit HTTPS Average Results

4.4.2.2 Latency and Jitter

Table 4.36 shows results consistent with previous findings. With jitter values non-existent or so low, they will have no effect, it can be concluded that the simulated traffic generated has no

direct or negative affect of the VPN performance.

Metric	Run 1	Run 2	Run 3	Run 4	Run 5
Latency	41ms	41ms	41ms	41ms	41ms
Jitter	0ms	0ms	0ms	0ms	0ms

Table 4.36: 8Mb SSL VPN AES 128-bit HTTPS Latency and Jitter Results

4.4.2.3 Packet Overhead

As with previous findings, packet headers have remained constant between all 8Mb SSL VPN tests, as can be seen in Table 4.37.

Attribute	Size (bytes)
IP Header	20
UDP Header	8
Security Layer	41
Avg. Packet Size	392.17

Table 4.37: 8Mb SSL VPN AES 128-bit HTTPS Packet Overhead Results

Table 4.37 shows an average packet size similar to that of the HTTP result. However, the value difference is so small it can be considered negligible and has no direct effect on VPN performance.

4.4.3 SSH

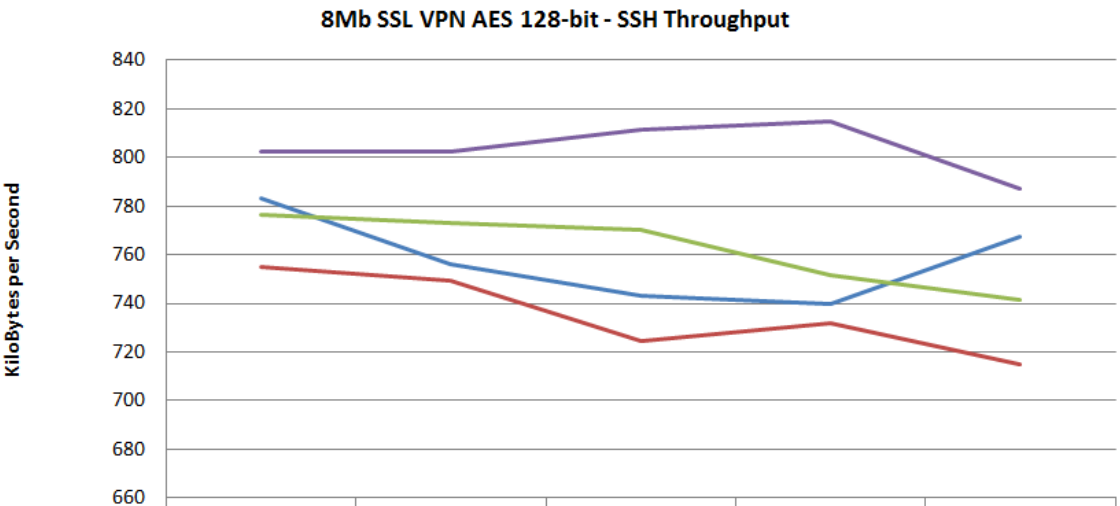
4.4.3.1 Throughput

As determined during the Blowfish and PPTP SSH results, it is expected that SSH performance would drop by around a margin of 5% compared to relative HTTP and HTTPS results. This is evident in 4.12 as it shows all transfer throughputs have dropped by roughly 5%. This result is indicative of a developing pattern, and it can be concluded that the process of encryption used by SSH incurs a noticeable performance drop of around 5% when compared to HTTP and HTTPS transfers.

Table 4.38 shows that the average throughput for each transfer has also dropped when compared against Blowfish SSH results. This would indicate that the AES 128-bit cipher is causing additional processing delay on every packet, amounting to a performance drop between 5-8% – but still within the acceptable 15% margin required to be cost-efficient whilst maintaining appropriate performance characteristics.

File Type	Avg. Throughput (KB/s)	Avg. Utilisation (%)
Video	757.94	74.02
Photo	735.12	71.79
Audio	762.60	74.47
Compressed	803.38	78.46

Table 4.38: 8Mb SSL VPN AES 128-bit SSH Average Results



	1	2	3	4	5
Video	783.36	755.89	743.37	739.68	767.41
Photo	755.12	749.09	724.66	731.75	714.98
Audio	776.44	773.19	770.30	751.38	741.70
Compressed	802.17	802.14	811.14	814.53	786.94

Figure 4.12: 8Mb SSL VPN AES 128-bit SSH Results

4.4.3.2 Latency and Jitter

Table 4.39 shows results consistent with previous findings. With jitter values non-existent or so low, they will have no effect, it can be concluded that the simulated traffic generated has no direct or negative affect of the VPN performance.

Metric	**Run 1**	**Run 2**	**Run 3**	**Run 4**	**Run 5**
Latency	41ms	41ms	41ms	41ms	41ms
Jitter	0ms	0ms	0ms	0ms	0ms

Table 4.39: 8Mb SSL VPN AES 128-bit SSH Latency and Jitter Results

4.4.3.3 Packet Overhead

As with previous findings, packet headers have remained constant between all 8Mb SSL VPN tests, as can be seen in Table 4.40.

Attribute	**Size (bytes)**
IP Header	20
UDP Header	8
Security Layer	41
Avg. Packet Size	374.33

Table 4.40: 8Mb SSL VPN AES 128-bit SSH Packet Overhead Results

Table 4.40 shows a lower average packet size compared to previous HTTP and HTTPS findings. This would suggest that SSH incurs further performance degradation, most likely due to additional overheads inside the frame payload, or an inefficient cipher being employed within SSH.

4.4.4 Conclusions

Tables 4.41 shows the average performance metrics of the AES 128-bit cipher, and indicates a relatively high-performing VPN. When compared to Table 4.42 which displays the findings from the Blowfish tests, a performance difference amounting to around 5% can be clearly seen.

Protocol	**Throughput (KB/s)**	**Utilisation (%)**
Video	816.32	79.72
Photo	775.39	75.72
Audio	797.21	77.85
Compressed	795.14	77.65

Table 4.41: 8Mb SSL VPN AES 128-bit Per File Type Summary Results

Protocol	**Throughput (KB/s)**	**Utilisation (%)**
Video	837.49	81.79
Photo	825.14	80.58
Audio	852.36	83.24
Compressed	887.14	86.64

Table 4.42: 8Mb SSL VPN Blowfish Per File Type Summary Results

For each file type transferred, the utilisation between the two ciphers is typically within a 5% margin. However, the compressed file appears to have a performance gap of around 10% – not consistent with previous findings. As no other results indicate a performance gap of this size, it can be considered an anomaly.

It is clear than the higher grade AES cipher incurs some performance drop when compared to the similar sized, but less advanced Blowfish cipher. Nevertheless, HTTP and HTTPS are still achieving very similar results and do not suggest a performance drop between protocols are theorised in Section 1.2.6.

Overall, the AES 128-bit cipher is able to achieve an average utilisation rate across all protocols of 77.74% – easily within the 15% margin when using the PPTP value of 85% as a baseline. This would suggest that several of the hypotheses stated in 1.2.6 are able to be validated.

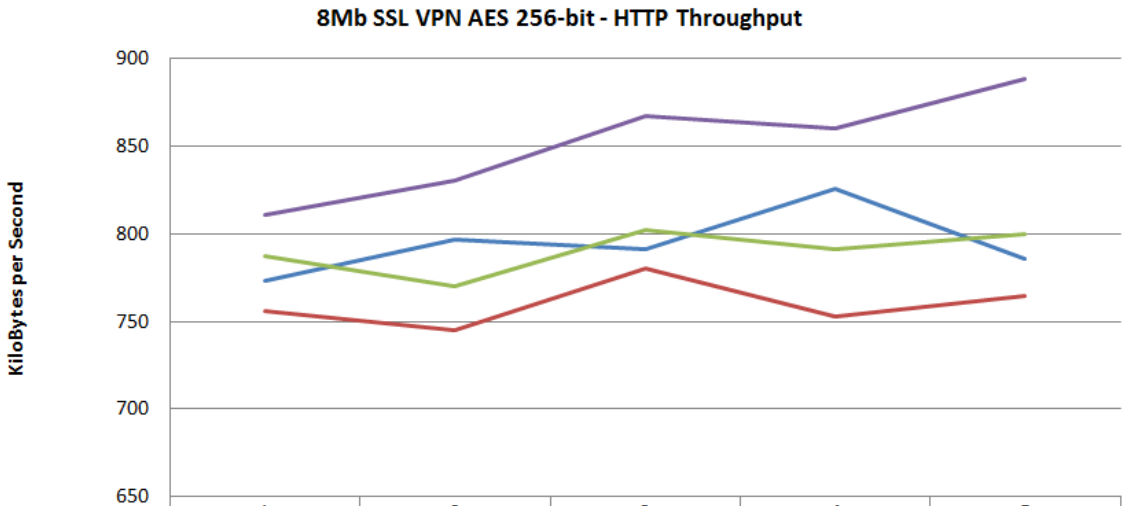
4.5 SSL VPN - 8Mb (1024 KB/s) - AES 256-bit

4.5.1 HTTP

4.5.1.1 Throughput

AES 256-bit is the strong cipher available in OpenVPN, as well as the only open cipher certified by the NSA for use in transmitting secret information (see Section 2.2.2.2. With the double key size, it is expected that the cipher will induce further CPU computation, thus, further degrading the performance of the VPN. However, Figure 4.13 shows that this statement may not be entirely correct.

Table 4.43 shows the compressed file, one again, achieving the fastest throughput, followed by



	1	2	3	4	5
Video	773.49	796.68	791.30	825.73	786.15
Photo	756.36	745.05	779.94	753.12	764.78
Audio	787.65	769.81	802.38	791.53	799.92
Compressed	811.1	830.44	867.60	860.34	888.69

Figure 4.13: 8Mb SSL VPN AES 256-bit HTTP Results

audio, video, and then photo – representative of earlier findings with audio and video achieving similar values throughout all tests conducted. However, average throughput and utilisation values have only dropped by roughly 3% compared to the AES 128-bit results – indicating that the increased key size does not have a noticeable detrimental effect on VPN performance.

File Type	**Avg. Throughput (KB/s)**	**Avg. Utilisation (%)**
Video	794.67	77.60
Photo	759.85	74.20
Audio	790.26	77.17
Compressed	851.63	83.17

Table 4.43: 8Mb SSL VPN AES 256-bit HTTP Average Results

4.5.1.2 Latency and Jitter

Table 4.44 shows results consistent with previous findings. With jitter values non-existent or so low, they will have no effect, it can be concluded that the simulated traffic generated has no direct or negative affect of the VPN performance.

Metric	**Run 1**	**Run 2**	**Run 3**	**Run 4**	**Run 5**
Latency	41ms	41ms	41ms	41ms	41ms
Jitter	0ms	0ms	0ms	0ms	0ms

Table 4.44: 8Mb SSL VPN AES 256-bit HTTP Latency and Jitter Results

4.5.1.3 Packet Overhead

As with previous findings, packet headers have remained constant between all 8Mb SSL VPN tests, as can be seen in Table 4.45.

Attribute	Size (bytes)
IP Header	20
UDP Header	8
Security Layer	41
Avg. Packet Size	392.54

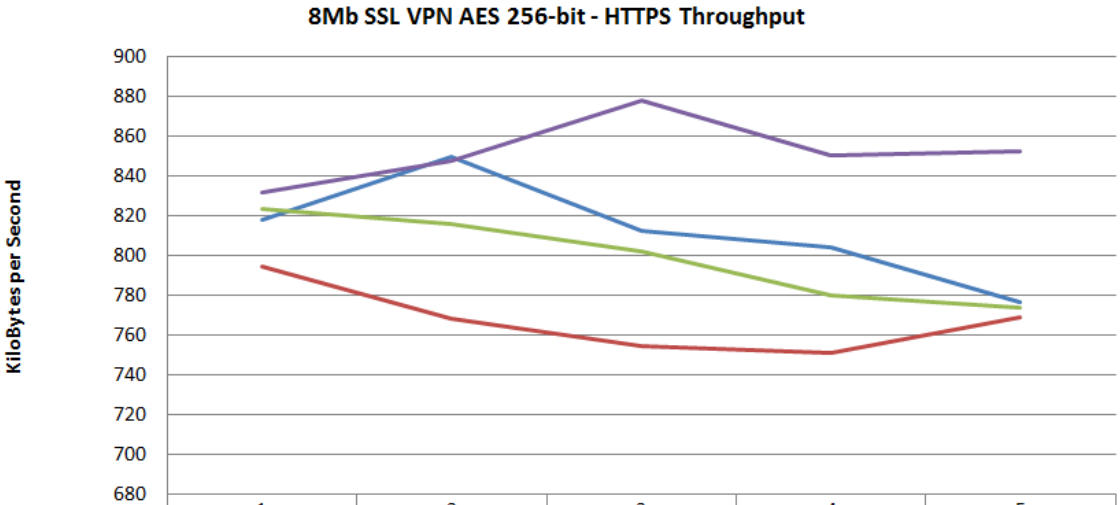
Table 4.45: 8Mb SSL VPN AES 256-bit HTTP Packet Overhead Results

Table 4.45 shows an average packet size consistent with AES 128-bit HTTP and HTTPS finding. This would suggest that the average packet sizing pattern (i.e. HTTP with the highest average packet size followed by HTTPS, then SSH), although noticeably consistent and deterministic, causes no discernable performance degradation.

4.5.2 HTTPS

4.5.2.1 Throughput

As with previous finding, it is expected that the HTTPS figures for AES 256-bit will be similar to earlier HTTP results. Figure 4.14 clearly shows that this statement to be true as most transfers actually had increase throughput and utilisation values compared to the earlier HTTP results.



	1	2	3	4	5
Video	817.93	849.60	812.38	804.25	776.88
Photo	794.44	768.80	754.34	751.36	769.40
Audio	823.47	815.74	802.35	779.93	773.79
Compressed	832.01	847.74	878.12	850.78	852.56

Figure 4.14: 8Mb SSL VPN AES 256-bit HTTPS Results

Table 4.46 shows the compressed file achieving the fastest throughput, and photo the slowest,

whilst audio and video are placed between the two. As expected, the HTTPS results were indicative of previous findings that suggested the performance would be relatively unaffected compared to HTTP transfers, as shown by the marginal increase of transfers – all within around a 2% margin of the HTTP results.

File Type	Avg. Throughput (KB/s)	Avg. Utilisation (%)
Video	812.21	79.32
Photo	767.67	74.97
Audio	799.06	78.03
Compressed	852.24	83.23

Table 4.46: 8Mb SSL VPN AES 256-bit HTTPS Average Results

4.5.2.2 Latency and Jitter

Table 4.47 shows results consistent with previous findings. With jitter values non-existent or so low, they will have no effect, it can be concluded that the simulated traffic generated has no direct or negative affect of the VPN performance.

Metric	Run 1	Run 2	Run 3	Run 4	Run 5
Latency	41ms	41ms	41ms	41ms	41ms
Jitter	0ms	0ms	0ms	0ms	0ms

Table 4.47: 8Mb SSL VPN AES 256-bit HTTPS Latency and Jitter Results

4.5.2.3 Packet Overhead

As with previous findings, packet headers have remained constant between all 8Mb SSL VPN tests, as can be seen in Table 4.48.

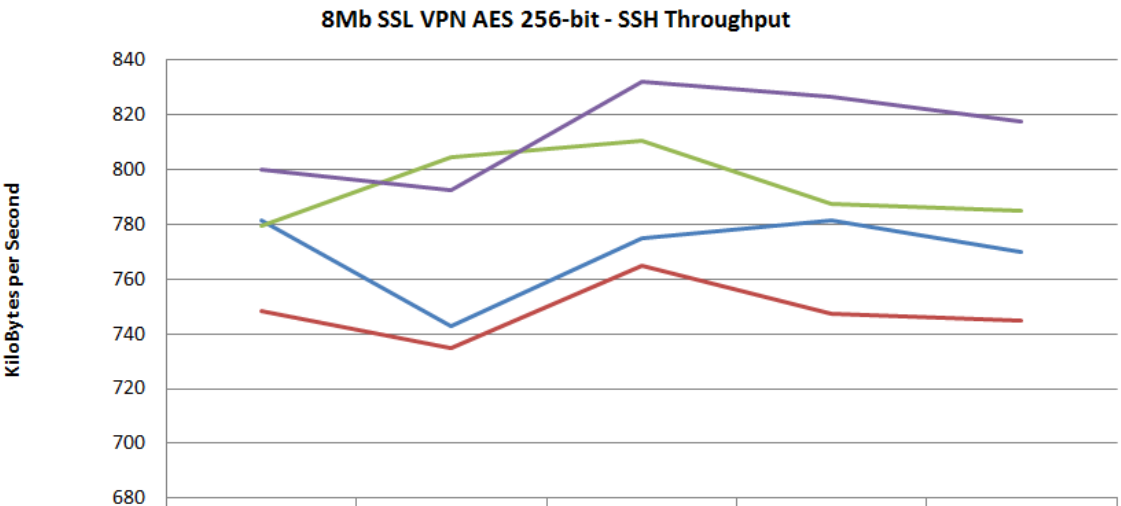
Attribute	Size (bytes)
IP Header	20
UDP Header	8
Security Layer	41
Avg. Packet Size	383.47

Table 4.48: 8Mb SSL VPN AES 256-bit HTTPS Packet Overhead Results

Table 4.48 shows an average packet size consistent with AES 128-bit HTTP and HTTPS finding. This would suggest that the average packet sizing pattern (i.e. HTTP with the highest average packet size followed by HTTPS, then SSH), although noticeably consistent and deterministic, causes no discernable performance degradation.

4.5.3 SSH

It is expected that the SSH figures for AES 256-bit will be similar to earlier AES 128-bit SSH results. Figure 4.15 shows this statement to be true as most transfers sit within a margin 2-3%, indicating no noticeable performance degradation.



	1	2	3	4	5
Video	781.69	742.86	774.99	781.65	769.99
Photo	748.22	735.02	765.04	747.24	744.67
Audio	779.35	804.32	810.34	787.27	785.19
Compressed	800.13	792.74	831.94	826.69	817.52

Figure 4.15: 8Mb SSL VPN AES 256-bit SSH Results

Table 4.49 shows results consistent with previous findings, and display the compressed file achieving the fastest throughput, followed by audio, video, and then photo. As expected, the SSH results were indicative of previous findings that suggested the performance would be less efficient compared to HTTP and HTTPS transfers, as shown by the performance decrease margin of around 5%.

File Type	**Avg. Throughput (KB/s)**	**Avg. Utilisation (%)**
Video	770.24	75.22
Photo	748.04	73.05
Audio	793.29	77.47
Compressed	813.80	79.47

Table 4.49: 8Mb SSL VPN AES 256-bit SSH Average Results

4.5.3.1 Latency and Jitter

Table 4.50 shows results consistent with previous findings. With jitter values non-existent or so low, they will have no effect, it can be concluded that the simulated traffic generated has no direct or negative affect of the VPN performance.

Metric	**Run 1**	**Run 2**	**Run 3**	**Run 4**	**Run 5**
Latency	41ms	41ms	41ms	41ms	41ms
Jitter	0ms	0ms	0ms	0ms	0ms

Table 4.50: 8Mb SSL VPN AES 256-bit SSH Latency and Jitter Results

4.5.3.2 Packet Overhead

As with previous findings, packet headers have remained constant between all 8Mb SSL VPN tests, as can be seen in Table 4.51.

Attribute	Size (bytes)
IP Header	20
UDP Header	8
Security Layer	41
Avg. Packet Size	357.19

Table 4.51: 8Mb SSL VPN AES 256-bit SSH Packet Overhead Results

Table 4.51 shows an average packet size consistent with AES 128-bit SSH findings, where the average packet size for SSH transfers are lower than both HTTP and HTTPS transfers. This could be attributed to the increased overheads occurred with SSH inside the frame payload, or just the inefficiencies of the SSH protocol compared to more modern encryption ciphers.

4.5.4 Conclusions

Tables 4.52 shows the average performance metrics of the AES 256-bit cipher, and indicates a relatively high-performing VPN. When compared to Table 4.53 which displays the findings from the AES 128-bit tests, a minute performance difference amounting to around 2-3% can be seen.

Protocol	Throughput (KB/s)	Utilisation (%)
Video	785.43	76.70
Photo	748.18	73.06
Audio	779.79	76.15
Compressed	809.07	79.01

Table 4.52: 8Mb SSL VPN AES 256-bit Per File Type Summary Results

Protocol	Throughput (KB/s)	Utilisation (%)
Video	837.49	81.79
Photo	825.14	80.58
Audio	852.36	83.24
Compressed	887.14	86.64

Table 4.53: 8Mb SSL VPN AES 128-bit Per File Type Summary Results

For each file type transferred, the utilisation between the two ciphers is typically within a relatively small margin of 2-3%. This would indicate that the extra computations required with the 256-bit key have no noticeable performance downsides to VPN performance, as the 2-3% margin is too small to consider significant.

It is clear than the higher grade 256-bit cipher incurs some performance drop when compared to its smaller 128-bit brother. Nevertheless, all protocols tested across both key sizes achieve very similar results and do not indicate an obvious performance drop.

Overall, the AES 256-bit cipher is still able to achieve relatively high-performance across all protocols, with an average utilisation of 76.23% – only 2% lower than AES 128-bit, and easily

within 15% PPTP baseline of 85%. This would suggest that several of the hypotheses stated in 1.2.6 are able to be validated.

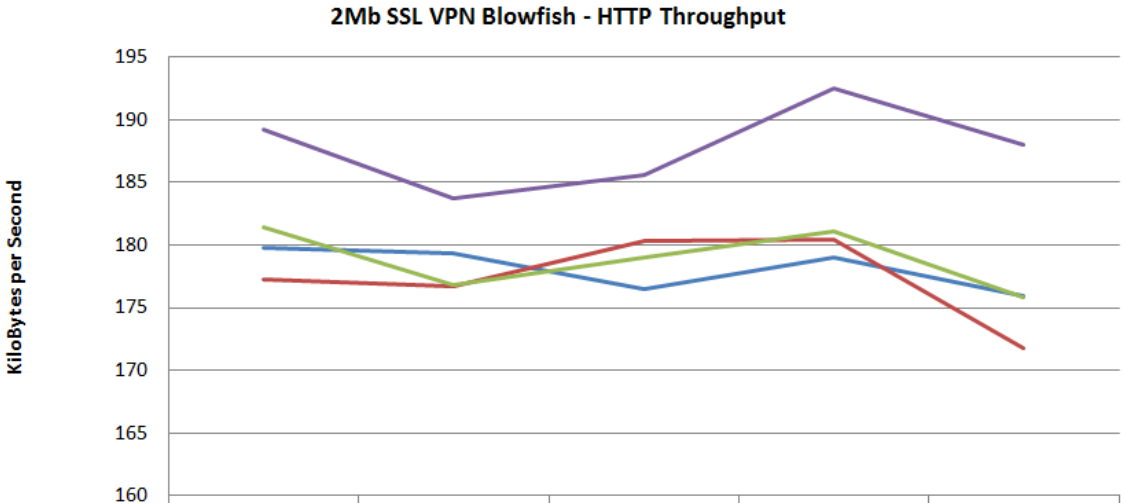
4.6 SSL VPN - 2Mb (256 KB/s) - Blowfish

As stated in Section 3.2.2, it was said by (Kuang, 2010) that increased latency will cause degradation in throughput. To test this theory, the SSL VPN tests will be run again with this increased latency – similar to PPTP. This will provide a reasonable and structured scenario where the results of both PPTP 8Mb and 2Mb tests can be compared against SSL VPN 8Mb and 2Mb tests.

4.6.1 HTTP

4.6.1.1 Throughput

As can be seen in Figure 4.16, the order of file transfers has remained fairly constant, with the compressed file transferring quickest, followed by audio, video and photo. Although some deviation in results occur, it can be concluded that the findings display a consistent and stable network due to the relatively minor fluctuations for each iteration of the file transfers.



	1	2	3	4	5
Video	179.74	179.34	176.48	179.04	175.92
Photo	177.29	176.75	180.36	180.46	171.79
Audio	181.39	176.84	179.02	181.12	175.84
Compressed	189.17	183.73	185.62	192.57	188.06

Figure 4.16: 2Mb SSL VPN Blowfish HTTP Results

The average throughput and utilisation values recorded in Table 4.54 clearly show a drop in performance across all transfers compared to its respective 8Mb test. Most noticeable is the 18% drop in performance during the audio transfer, as well as an overall performance drop margin of around 14-15%. This would correlate relatively well to the 2Mb PPTP results were a performance drop of 20% was observed compared to its respective 8Mb tests. It would also

indicate that most 2Mb results will follow this performance drop, which would result in SSL VPNs obtaining a performance level below 70% utilisation – disproving some hypotheses in Section 1.2.6.

File Type	Avg. Throughput (KB/s)	Avg. Utilisation (%)
Video	178.10	69.57
Photo	177.33	69.27
Audio	178.84	69.86
Compressed	187.83	73.37

Table 4.54: 2Mb SSL VPN Blowfish HTTP Average Results

Currently Table 4.54 display the majority of the transfers below the specified 70% utilisation. This would suggest that results from more robust and secure, but slightly slower ciphers, will not achieve an acceptable performance level.

4.6.1.2 Latency and Jitter

Table 4.55 shows an increased latency (as expected) compared to previous 8Mb SSL VPN tests. The latency values also correlate and confirm that values attained during the 2Mb PPTP tests. With jitter values non-existent or so low, they will have no effect, it can be concluded that the simulated traffic generated has no direct or negative affect of the VPN performance. However, the higher latency times appear to affect throughput performance, but further tests are required to confirm.

Metric	Run 1	Run 2	Run 3	Run 4	Run 5
Latency	115ms	115ms	115ms	115ms	115ms
Jitter	0ms	0ms	0ms	0ms	0ms

Table 4.55: 2Mb SSL VPN Blowfish HTTP Latency and Jitter Results

4.6.1.3 Packet Overhead

Packet headers have remained constant across all SSL VPN tests, as can be seen in Table 4.56.

Attribute	Size (bytes)
IP Header	20
UDP Header	8
Security Layer	41
Avg. Packet Size	399.48

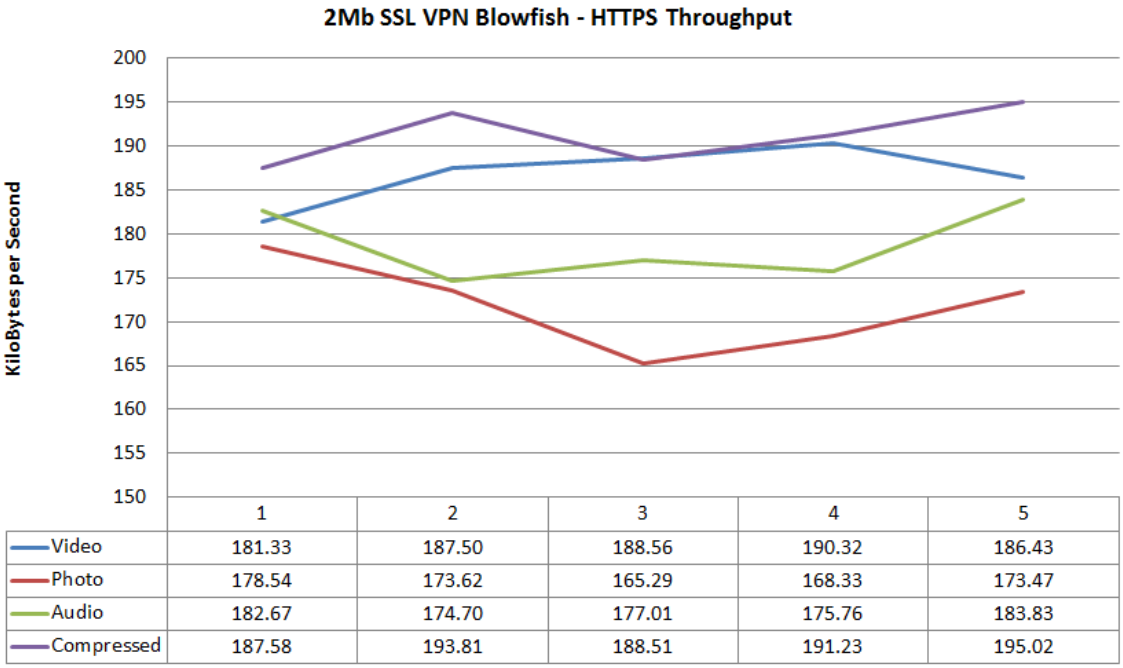
Table 4.56: 2Mb SSL VPN Blowfish HTTP Packet Overhead Results

Table 4.56 shows an average packet size consistent with previous 8Mb SSL VPN tests. This would suggest that the average packet size causes no discernable performance degradation.

4.6.2 HTTPS

4.6.2.1 Throughput

Based on previous HTTPS finding, it is expected that the difference between the 2Mb SSL VPN with Blowfish HTTP and HTTPS results will be relatively small. Figure 4.17 proves this statement as all values sit within a margin of approximately 3% compared to HTTP findings. These results are consistent with earlier findings that there is no discernable performance difference whilst using the HTTP and HTTPS protocol.



	1	2	3	4	5
Video	181.33	187.50	188.56	190.32	186.43
Photo	178.54	173.62	165.29	168.33	173.47
Audio	182.67	174.70	177.01	175.76	183.83
Compressed	187.58	193.81	188.51	191.23	195.02

Figure 4.17: 2Mb SSL VPN Blowfish HTTPS Results

Table 4.57 indicates a performance drop compared to the 8Mb SSL VPN Blowfish HTTPS test. With an average performance discrepancy of 15%, this is consistent with the 2Mb PPTP performance findings, and seems to indicate more clearly that increased latency does affect throughput speeds. Also noticeable are the utilisation values of video, photo and audio, which all sit relatively close to the 70% performance level required to consider SSL VPNs a cost-efficient and adequate alternative to traditional PPTP VPNs.

File Type	**Avg. Throughput (KB/s)**	**Avg. Utilisation (%)**
Video	186.83	72.98
Photo	171.85	67.13
Audio	178.79	69.84
Compressed	191.23	74.69

Table 4.57: 2Mb SSL VPN Blowfish HTTPS Average Results

4.6.2.2 Latency and Jitter

Table 4.58 shows a latency value consistent with the HTTP test. With jitter values non-existent or so low, they will have no effect, it can be concluded that the simulated traffic generated has no direct or negative affect of the VPN performance. The increase latency times still appear to affect throughput performance, but all protocols and ciphers must first be tested before comprising a final conclusion.

Metric	**Run 1**	**Run 2**	**Run 3**	**Run 4**	**Run 5**
Latency	115ms	115ms	115ms	115ms	115ms
Jitter	0ms	0ms	0ms	0ms	0ms

Table 4.58: 2Mb SSL VPN Blowfish HTTPS Latency and Jitter Results

4.6.2.3 Packet Overhead

Packet headers have remained constant across all SSL VPN tests, as can be seen in Table 4.59.

Attribute	**Size (bytes)**
IP Header	20
UDP Header	8
Security Layer	41
Avg. Packet Size	383.67

Table 4.59: 2Mb SSL VPN Blowfish HTTPS Packet Overhead Results

Table 4.59 shows an average packet size consistent with previous SSL VPN tests. This would suggest that the average packet size causes no discernable performance degradation.

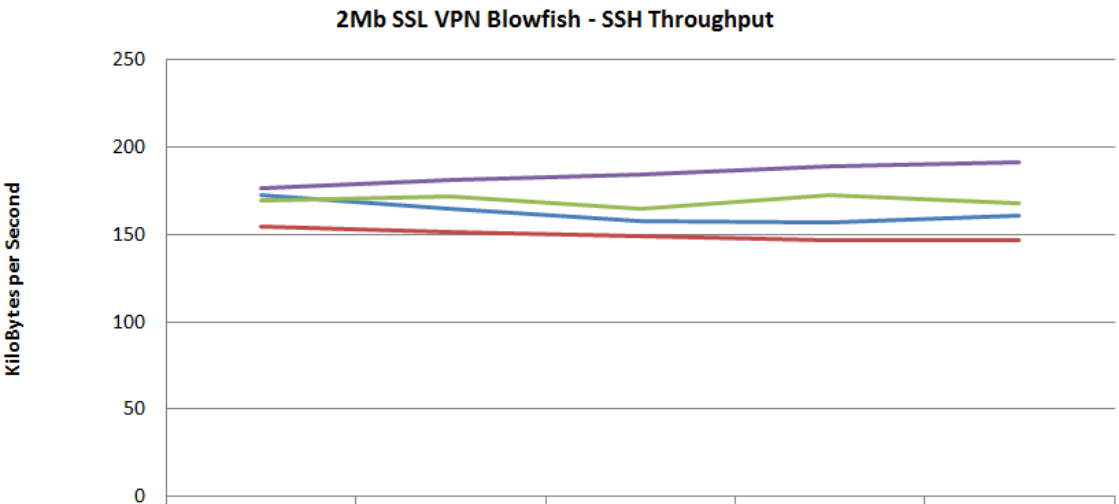
4.6.3 SSH

4.6.3.1 Throughput

It is expected that the SSH figures will be consistent with previous SSH findings where there is a further degradation in performance of roughly 5% compared to relative HTTP and HTTPS figures. This can be seen in Figure 4.18 where this statement holds true. As can be clearly seen, all iterations of each transfer were relatively similar, thus, resulting in what appears to be almost straight lines – indicative of a stable and consistent network.

Table 4.60 shows results consistent with previous findings, and displays the compressed file achieving the fastest throughput, followed by audio, video, and then photo. As expected, the SSH results were indicative of previous findings that suggested the performance would be less efficient compared to HTTP and HTTPS transfers, as shown by the performance decrease margin of around 5%.

However, the utilisation values are now decreasing with each protocol and higher grade cipher in use, and it could be speculated that most values (which already fall outside the 70% acceptable performance level) will soon approach close to the 50% utilisation value obtained by the (Khanvilkar & Khokhar, 2004) experiment.



	1	2	3	4	5
Video	172.47	164.80	158.15	157.17	161.26
Photo	154.68	151.60	149.10	146.84	146.48
Audio	169.63	172.07	165.19	172.66	168.12
Compressed	176.73	181.67	184.36	189.41	191.24

Figure 4.18: 2Mb SSL VPN Blowfish SSH Results

File Type	Avg. Throughput (KB/s)	Avg. Utilisation (%)
Video	162.77	63.58
Photo	149.74	58.49
Audio	169.53	66.22
Compressed	184.68	72.14

Table 4.60: 2Mb SSL VPN Blowfish SSH Average Results

4.6.3.2 Latency and Jitter

Table 4.61 shows results consistent with previous findings. With jitter values non-existent or so low, they will have no effect, it can be concluded that the simulated traffic generated has no direct or negative affect of the VPN performance. Once again, the increased latency appears to have affected the throughput; however, it is more likely that a number of factors contribute to the decreased performance such as the use of the less efficient SSH protocol.

Metric	Run 1	Run 2	Run 3	Run 4	Run 5
Latency	115ms	115ms	115ms	115ms	115ms
Jitter	0ms	0ms	0ms	0ms	0ms

Table 4.61: 2Mb SSL VPN Blowfish SSH Latency and Jitter Results

4.6.3.3 Packet Overhead

As with previous findings, packet headers have remained constant between all SSL VPN tests, as can be seen in Table 4.62.

Table 4.62 shows an average packet size consistent with previous SSL findings, where the average

Attribute	Size (bytes)
IP Header	20
UDP Header	8
Security Layer	41
Avg. Packet Size	342.76

Table 4.62: 2Mb SSL VPN Blowfish SSH Packet Overhead Results

packet size for SSH transfers are lower than both HTTP and HTTPS transfers. This could be attributed to the increased overheads occurred with SSH inside the frame payload, or just the inefficiencies of the SSH protocol compared to more modern encryption ciphers.

4.6.4 Conclusions

Tables 4.63 shows the average performance metrics of the 2Mb SSL VPN Blowfish test, and indicates a reasonable performance level averaging at 70%. When compared to Table 4.64, which displays the findings from the 8Mb Blowfish tests, a noticeable performance difference amounting to around 15% can be seen.

Protocol	Throughput (KB/s)	Utilisation (%)
Video	179.46	70.10
Photo	169.39	66.17
Audio	178.69	69.80
Compressed	186.98	73.04

Table 4.63: 2Mb SSL VPN Blowfish Per File Type Summary Results

Protocol	Throughput (KB/s)	Utilisation (%)
Video	837.49	81.79
Photo	825.14	80.58
Audio	852.36	83.24
Compressed	887.14	86.64

Table 4.64: 8Mb SSL VPN Blowfish Per File Type Summary Results

These results would indicate that the difference between similar tests, but different latencies, results in around a 15% performance drop. This would support the conclusion by (Kuang, 2010) that increased latency does have a direct effect on throughput performance. However, further tests comparing other ciphers are required to confirm.

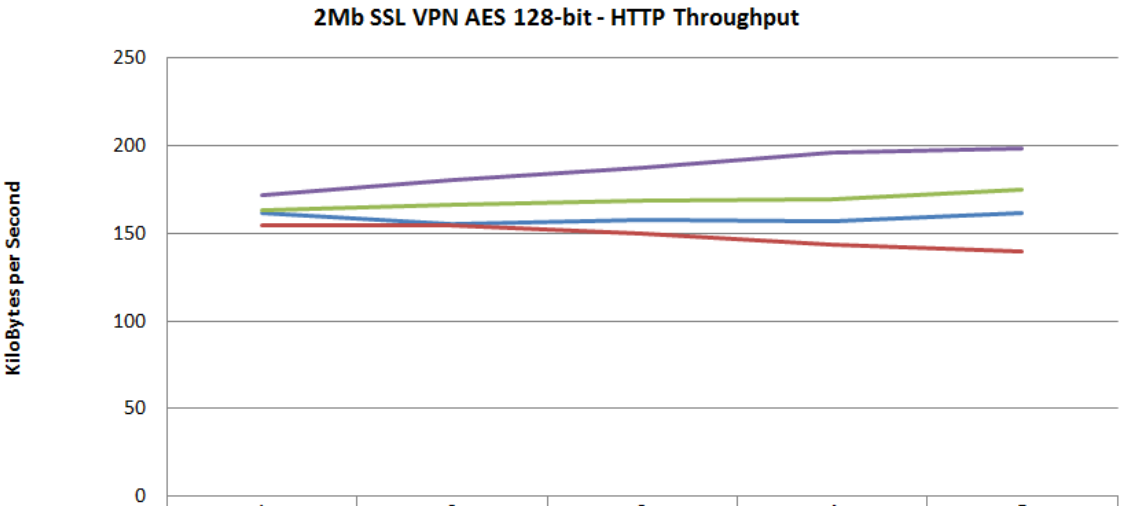
Overall, the Blowfish cipher is still able to achieve moderate performance across all protocols, with an average utilisation of 69.78% – just outside the 15% margin of the PPTP baseline of 85%. This would suggest higher grade ciphers would incur further performance degradation, thus, not achieving adequate performance levels to support the use of SSL VPNs on low-bandwidth and high latency links.

4.7 SSL VPN - 2Mb (256 KB/s) - AES 128-bit

4.7.1 HTTP

4.7.1.1 Throughput

As can be seen in Figure 4.19, the order of file transfers has remained fairly constant, with the compressed file transferring quickest, followed by audio, video and photo. Although some deviation in results occur, it can be concluded that the findings display a consistent and stable network due to the relatively minor fluctuations for each iteration of the file transfers.



	1	2	3	4	5
Video	161.47	155.78	158.11	157.05	162.08
Photo	154.63	154.58	150.21	143.48	139.92
Audio	163.33	166.29	168.86	169.40	175.21
Compressed	172.02	180.37	187.19	195.81	198.81

Figure 4.19: 2Mb SSL VPN AES 128-bit HTTP Results

The average throughput and utilisation values recorded in Table 4.65 clearly show a drop in performance across all transfers compared to its respective 8Mb test. Most noticeable is the 18% drop in performance during the video transfer, as well as an overall performance drop margin of over 15%. This would correlate relatively well to the 2Mb PPTP results where a performance drop of 20% was observed compared to its respective 8Mb tests.

File Type	Avg. Throughput (KB/s)	Avg. Utilisation (%)
Video	158.90	62.07
Photo	148.56	58.03
Audio	168.62	65.87
Compressed	186.84	72.98

Table 4.65: 2Mb SSL VPN AES 128-bit HTTP Average Results

Currently Table 4.65 displays the majority of the transfers below the specified 70% utilisation. This would suggest that AES 128-bit is not suitable for use on slow-speed, high latency links. It would also be reasonable to speculate that future AES 256-bit results would fall short of the 70% margin required to constitute an adequate performing VPN suitable for small businesses.

4.7.1.2 Latency and Jitter

Table 4.66 shows latency values consistent with previous 2Mb tests where there is a constant and non-fluctuating value of 115ms across all iterations. The latency values also confirm values attained during the 2Mb PPTP tests. With jitter values non-existent or so low, they will have no effect, it can be concluded that the simulated traffic generated has no direct or negative affect of the VPN performance.

Metric	Run 1	Run 2	Run 3	Run 4	Run 5
Latency	115ms	115ms	115ms	115ms	115ms
Jitter	0ms	0ms	0ms	0ms	0ms

Table 4.66: 2Mb SSL VPN AES 128-bit HTTP Latency and Jitter Results

4.7.1.3 Packet Overhead

Packet headers have remained constant across all SSL VPN tests, as can be seen in Table 4.67.

Attribute	Size (bytes)
IP Header	20
UDP Header	8
Security Layer	41
Avg. Packet Size	402.75

Table 4.67: 2Mb SSL VPN AES 128-bit HTTP Packet Overhead Results

Table 4.67 shows an average packet size consistent with previous SSL VPN tests where all values were found to be in the range of 385-405 bytes. This would suggest that the average packet size causes no discernable performance degradation.

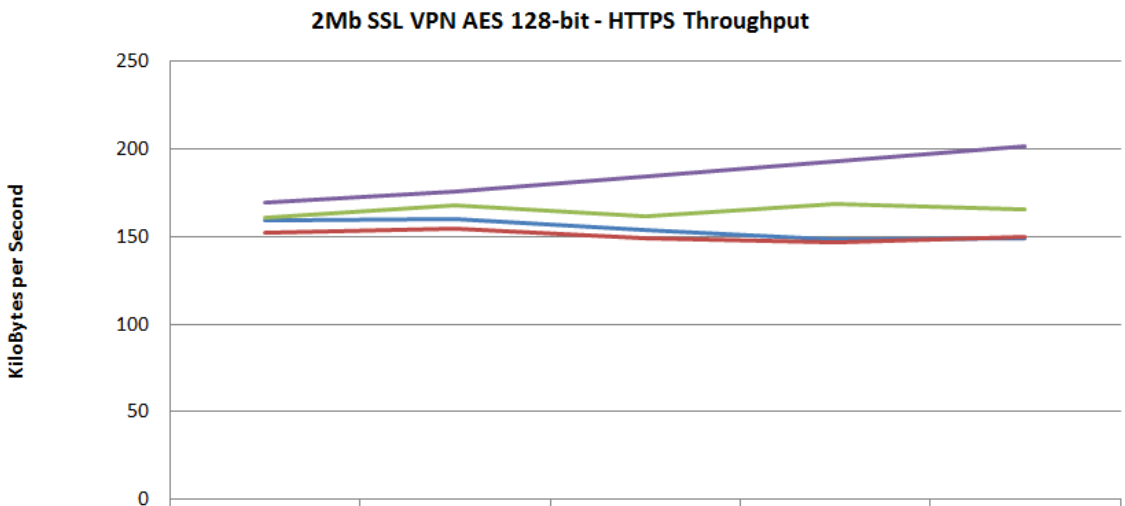
4.7.2 HTTPS

4.7.2.1 Throughput

Following previously established patterns, it is expected that the HTTPS values will be consistent with HTTP values and fall within a margin of 2-3%. This is evident in Figure 4.20 where the order of file transfers has remained fairly constant, with the compressed file transferring quickest, followed by audio, video and photo. Also, the shape of each line in the graph also represents consistent findings when compared against HTTP results – indicative of a stable network.

Table 4.68 clearly shows no discernable difference in performance compared to its relative HTTP test – as expected. However, compared against its 8Mb equivalent, the values correlate to an average drop of around 15%. This would support previous speculations that high latency links can cause a 15% drop in performance.

Once again, the majority of the transfers have returned values below the needed 70% utilisation, as can be seen in Table 4.68. Whilst fairly close to the required 70%, it would suggest that AES



	1	2	3	4	5
Video	159.36	160.25	154.13	148.74	149.22
Photo	152.48	154.36	149.38	146.77	149.66
Audio	161.14	167.79	162.01	168.46	166.00
Compressed	169.84	175.58	184.34	193.18	202.00

Figure 4.20: 2Mb SSL VPN AES 128-bit HTTPS Results

File Type	Avg. Throughput (KB/s)	Avg. Utilisation (%)
Video	154.34	60.29
Photo	150.53	58.80
Audio	165.08	64.48
Compressed	184.99	72.26

Table 4.68: 2Mb SSL VPN AES 128-bit HTTPS Average Results

128-bit is not suitable for use on slow-speed, high latency links, thus, is not adequate for small businesses unless average performance is acceptable.

4.7.2.2 Latency and Jitter

Similar to the HTTP results, Table 4.69 shows latency values consistent with those results with a constant and non-fluctuating value of 115ms. With jitter values non-existent or so low, they will have no effect, it can be concluded that the simulated traffic generated has no direct or negative affect of the VPN performance.

Metric	Run 1	Run 2	Run 3	Run 4	Run 5
Latency	115ms	115ms	115ms	115ms	115ms
Jitter	0ms	0ms	0ms	0ms	0ms

Table 4.69: 2Mb SSL VPN AES 128-bit HTTPS Latency and Jitter Results

4.7.2.3 Packet Overhead

Packet headers have remained constant across all SSL VPN tests, as can be seen in Table 4.70.

Attribute	Size (bytes)
IP Header	20
UDP Header	8
Security Layer	41
Avg. Packet Size	397.62

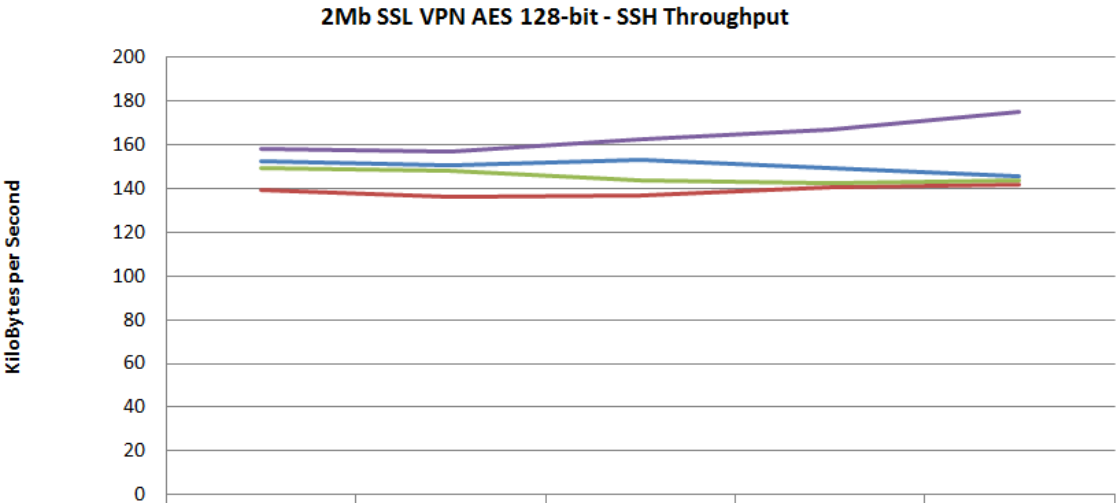
Table 4.70: 2Mb SSL VPN AES 128-bit HTTPS Packet Overhead Results

Table 4.70 shows an average packet size of 397.62 bytes, consistent with previous SSL VPN tests where all values were found to be in the range of 385-405 bytes. This would suggest that the average packet size causes no discernable performance degradation.

4.7.3 SSH

4.7.3.1 Throughput

Figure 4.21 displays consistent values across all 5 iterations for each transfer. As with previous tests, the compressed file had the highest throughput, followed by an almost inseparable video and audio values, and photo with the slowest throughput. From the shape of the lines for each file type, it can be concluded that the network is consistent and stable as it provides very similar results across most tests.



	1	2	3	4	5
Video	152.84	150.67	153.37	149.40	145.75
Photo	139.45	136.55	136.68	140.38	141.59
Audio	149.33	147.97	143.78	142.79	144.01
Compressed	158.12	156.96	162.32	166.87	175.10

Figure 4.21: 2Mb SSL VPN AES 128-bit SSH Results

As expected, based on 8Mb SSL VPN AES 128-bit SSH results, Table 4.71 clearly shows a difference in performance of around 15-20% – consistent with previous findings based on results between all other 8Mb and 2Mb tests. These values provide reliable evidence that the high latency links are causing performance drops of 15-20% across all protocols.

All values recorded in Table 4.71 are below the needed adequate performance level of 70%. This

File Type	Avg. Throughput (KB/s)	Avg. Utilisation (%)
Video	150.41	58.75
Photo	138.93	54.27
Audio	145.58	56.87
Compressed	163.87	64.01

Table 4.71: 2Mb SSL VPN AES 128-bit SSH Average Results

would suggest that AES 128-bit is not suitable for use on slow-speed, high latency links, thus, is not adequate for small businesses unless average performance is acceptable.

4.7.3.2 Latency and Jitter

Table 4.72 shows latency values consistent with previous 2Mb results of 115ms. With jitter values non-existent or so low, they will have no effect, it can be concluded that the simulated traffic generated has no direct or negative affect of the VPN performance.

Metric	Run 1	Run 2	Run 3	Run 4	Run 5
Latency	115ms	115ms	115ms	115ms	115ms
Jitter	0ms	0ms	0ms	0ms	0ms

Table 4.72: 2Mb SSL VPN AES 128-bit SSH Latency and Jitter Results

4.7.3.3 Packet Overhead

Packet headers have remained constant across all SSL VPN tests, as can be seen in Table 4.73.

Attribute	Size (bytes)
IP Header	20
UDP Header	8
Security Layer	41
Avg. Packet Size	389.92

Table 4.73: 2Mb SSL VPN AES 128-bit SSH Packet Overhead Results

Table 4.73 shows an average packet size of 389.92 bytes, consistent with previous SSL VPN tests where all values were found to be in the range of 385-405 bytes. This would suggest that the average packet size causes no discernable performance degradation.

4.7.4 Conclusions

Tables 4.74 shows the average performance metrics of the 2Mb SSL VPN AES 128-bit test, and indicates a reasonable performance level averaging at 61.22%. When compared to its relative 8Mb test shown in Table 4.75, a noticeable performance difference amounting to around 15% can be seen – consistent with previous findings between all 8Mb and 2Mb tests.

The average performance drop of around 15% would support (Kuang, 2010) in his statement that increased latency does have a direct effect on throughput performance. After testing two

Protocol	Throughput (KB/s)	Utilisation (%)
Video	162.48	63.47
Photo	141.652	55.33
Audio	157.02	61.33
Compressed	165.74	64.74

Table 4.74: 2Mb SSL VPN AES 128-bit Per File Type Summary Results

Protocol	Throughput (KB/s)	Utilisation (%)
Video	816.32	79.72
Photo	775.39	75.72
Audio	797.21	77.85
Compressed	795.14	77.65

Table 4.75: 8Mb SSL VPN AES 128-bit Per File Type Summary Results

VPN tunnelling methods and 3 ciphers, it would be fairly safe to conclude that this is indeed true.

Overall, the AES 128-bit cipher is still able to achieve moderate performance across all protocols, with an average utilisation of 61.22%. Although this value is now well outside the 70% performance level required. This would suggest that whilst higher grade ciphers provide increased security, they incur much lower performance over low-bandwidth and high latency links. Thus, may not be suitable for small businesses that require high performance VPNs over such links.

4.8 SSL VPN - 2Mb (256 KB/s) - AES 256-bit

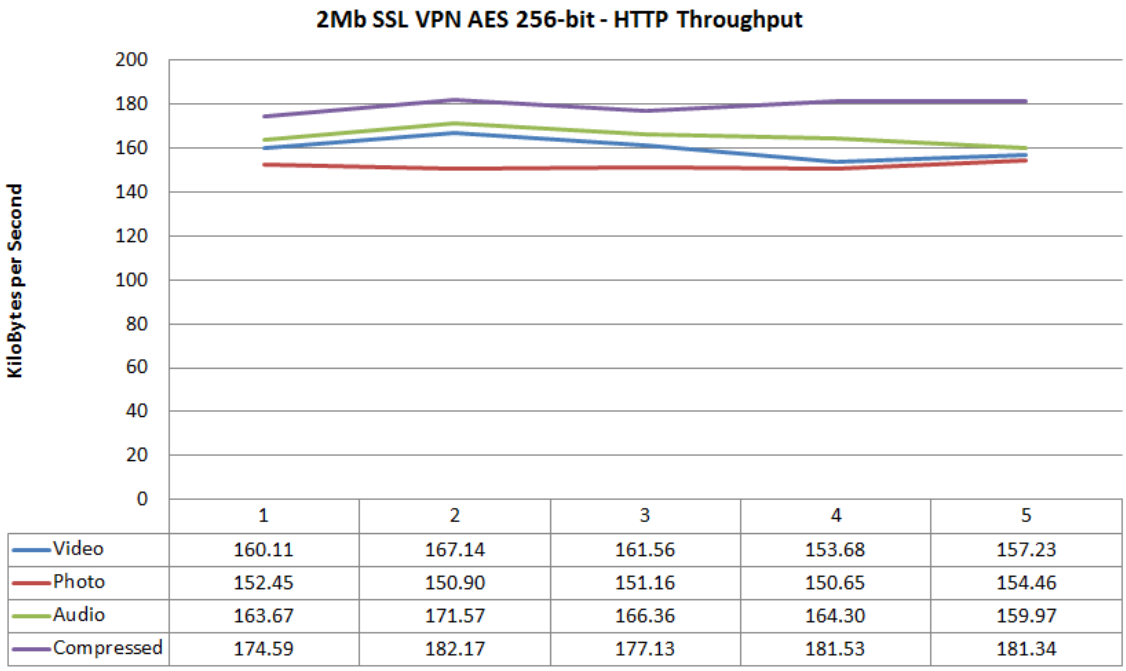
4.8.1 HTTP

4.8.1.1 Throughput

Based on previous results from the 8Mb AES 256-bit tests, it is expected that the increase in key size will not noticeably affect the overall performance. This can be seen in Figure 4.22, where the order of file transfer performance has remained fairly constant, with the compressed file transferring quickest, followed by audio, video and photo. Although some deviation in results occur, it can be concluded that the findings display a consistent and stable network due to the relatively minor fluctuations for each iteration of the file transfers.

As expected, Table 4.76 shows a distinct similarity in performance between AES 128-bit and 256-bit, as all transfers have a negligible 2-3% difference. Compared to the 8Mb SSL VPN AES 256-bit results, they are consistent with all other 2Mb results where performance has dropped across all protocols by around 15%. This would correlate to the previous findings that a performance drop of 15-20% is to be expected between 8Mb and 2Mb results – indicating that the higher latency is directly affecting the throughput.

Table 4.76 displays the majority of the transfers below the specified adequate performance level of 70% utilisation of the link speed. This would suggest that, as with AES 128-bit, that AES 256-bit is not suitable for use on slow-speed, high latency links.



	1	2	3	4	5
Video	160.11	167.14	161.56	153.68	157.23
Photo	152.45	150.90	151.16	150.65	154.46
Audio	163.67	171.57	166.36	164.30	159.97
Compressed	174.59	182.17	177.13	181.53	181.34

Figure 4.22: 2Mb SSL VPN AES 256-bit HTTP Results

File Type	Avg. Throughput (KB/s)	Avg. Utilisation (%)
Video	159.944	62.48
Photo	151.92	59.34
Audio	165.17	64.52
Compressed	179.35	70.06

Table 4.76: 2Mb SSL VPN AES 256-bit HTTP Average Results

4.8.1.2 Latency and Jitter

Table 4.77 shows latency values consistent with previous 2Mb tests where there is a constant and non-fluctuating value of 115ms across all iterations. The latency values also confirm values attained during the 2Mb PPTP tests. With jitter values non-existent or so low, they will have no effect, it can be concluded that the simulated traffic generated has no direct or negative affect of the VPN performance.

Metric	Run 1	Run 2	Run 3	Run 4	Run 5
Latency	115ms	115ms	115ms	115ms	115ms
Jitter	0ms	0ms	0ms	0ms	0ms

Table 4.77: 2Mb SSL VPN AES 256-bit HTTP Latency and Jitter Results

4.8.1.3 Packet Overhead

Packet headers have remained constant across all SSL VPN tests, as can be seen in Table 4.78.

Table 4.78 shows an average packet size consistent with previous SSL VPN tests where all values were found to be in the range of 385-405 bytes. This would suggest that the average packet size

Attribute	Size (bytes)
IP Header	20
UDP Header	8
Security Layer	41
Avg. Packet Size	399.83

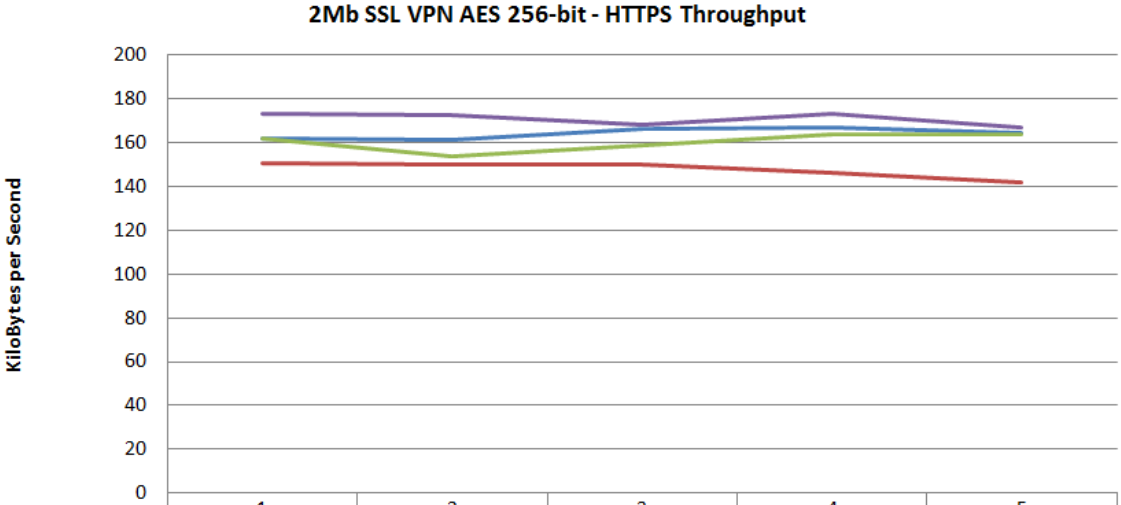
Table 4.78: 2Mb SSL VPN AES 256-bit HTTP Packet Overhead Results

causes no discernable performance degradation.

4.8.2 HTTPS

4.8.2.1 Throughput

Figure 4.23 shows a similar graph to the HTTP results, indicating consistent results across all HTTP and HTTPS tests where there is very little difference between HTTP and HTTPS findings. Once again, the order of file transfers has remained fairly constant, with the compressed file transferring quickest, followed by audio, video and photo. Also, the straight nature of each line representing each file type is indicative of a consistent and stable network.



	1	2	3	4	5
Video	162.13	161.57	166.27	166.95	164.57
Photo	150.56	149.84	149.93	146.60	142.07
Audio	161.97	153.89	158.88	163.95	164.15
Compressed	173.14	172.85	167.97	173.08	166.75

Figure 4.23: 2Mb SSL VPN AES 256-bit HTTPS Results

Table 4.79 clearly shows no discernable difference in performance compared to its relative HTTP test – as expected. However, compared against its 8Mb equivalent, the values correlate to an average drop of around 15%. This would support previous speculations that high latency links can cause an approximate 15-20% drop in performance. Each transfer was able to perform within a 2-3% margin of its relative HTTP results, except the compressed file. In this test, the compressed file averaged a 4% margin; however, this value is so close to the expected 2-3% that it can be disregarded as within a reasonable margin of error.

File Type	Avg. Throughput (KB/s)	Avg. Utilisation (%)
Video	164.29	64.18
Photo	147.80	57.73
Audio	160.57	62.72
Compressed	170.76	66.70

Table 4.79: 2Mb SSL VPN AES 256-bit HTTPS Average Results

All transfers have returned values below the needed 70% utilisation, as can be seen in Table 4.68. Whilst fairly close to the required 70%, it would suggest that AES 256-bit is not suitable for use on slow-speed, high latency links, thus, is not adequate for small businesses unless average performance is acceptable.

4.8.2.2 Latency and Jitter

Similar to the HTTP results, Table 4.80 shows latency values consistent with those results with a constant and non-fluctuating value of 115ms. With jitter values non-existent or so low, they will have no effect, it can be concluded that the simulated traffic generated has no direct or negative affect of the VPN performance.

Metric	Run 1	Run 2	Run 3	Run 4	Run 5
Latency	115ms	115ms	115ms	115ms	115ms
Jitter	0ms	0ms	0ms	0ms	0ms

Table 4.80: 2Mb SSL VPN AES 256-bit HTTPS Latency and Jitter Results

4.8.2.3 Packet Overhead

Packet headers have remained constant across all SSL VPN tests, as can be seen in Table 4.81.

Attribute	Size (bytes)
IP Header	20
UDP Header	8
Security Layer	41
Avg. Packet Size	392.94

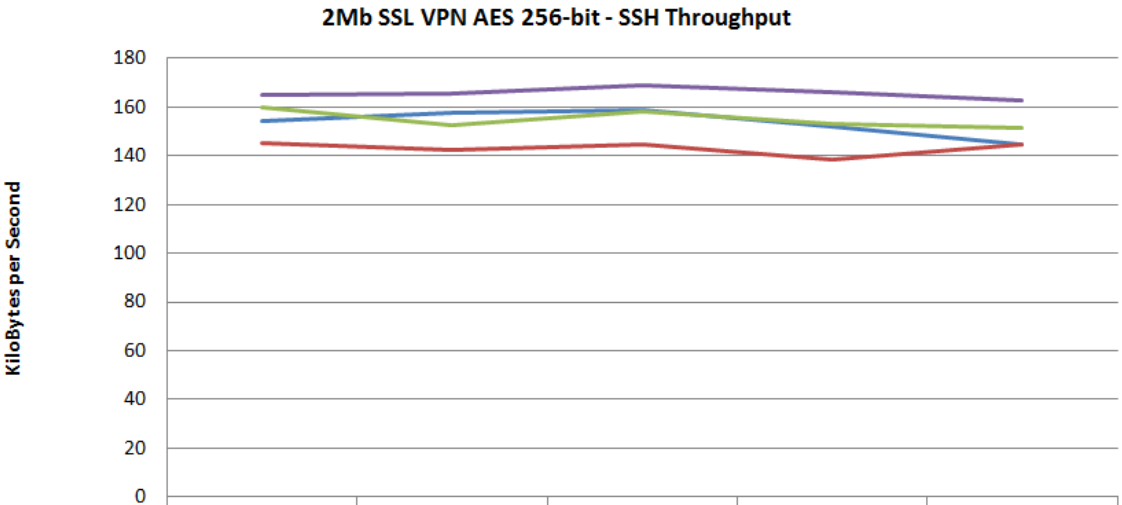
Table 4.81: 2Mb SSL VPN AES 256-bit HTTPS Packet Overhead Results

Table 4.81 shows an average packet size of 392.94 bytes, consistent with previous SSL VPN tests where all values were found to be in the range of 385-405 bytes. This would suggest that the average packet size causes no discernable performance degradation.

4.8.3 SSH

4.8.3.1 Throughput

Figure 4.24 displays consistent values across all 5 iterations for each transfer as indicated by the straight nature of each line. As with most other tests, the compressed file had the highest throughput, followed by audio, video, and photo with the slowest throughput.



	1	2	3	4	5
Video	154.39	157.42	158.99	152.02	144.82
Photo	145.25	142.65	144.89	138.19	144.52
Audio	159.73	152.81	158.29	153.35	151.31
Compressed	165.19	165.74	169.12	165.91	162.78

Figure 4.24: 2Mb SSL VPN AES 256-bit SSH Results

Table 4.82 clearly shows drop in performance of around 15-20% compared to its relative 8Mb test, as well as being consistent with all other drops in performance between 8Mb and 2Mb tests. These values provide reliable evidence that the high latency links are causing performance drops of 15-20% across all protocols. Also noticeable is the slight performance drop, indicative of SSH, of around 5% compared against the HTTP and HTTPS results. This is most likely due to additional overheads inside the frame payload or inefficiency in the SSH protocol.

File Type	**Avg. Throughput (KB/s)**	**Avg. Utilisation (%)**
Video	153.52	59.97
Photo	143.10	55.89
Audio	155.09	60.58
Compressed	165.75	64.75

Table 4.82: 2Mb SSL VPN AES 256-bit SSH Average Results

Table 4.82 once again shows values below the needed adequate performance level of 70%. This would suggest that AES 256-bit is definitively not suitable for use on slow-speed, high latency links, thus, is not adequate for small businesses unless average performance is acceptable.

4.8.3.2 Latency and Jitter

Table 4.83 shows latency values consistent with previous 2Mb results of 115ms. With jitter values non-existent or so low, they will have no effect, it can be concluded that the simulated traffic generated has no direct or negative affect of the VPN performance.

Metric	Run 1	Run 2	Run 3	Run 4	Run 5
Latency	115ms	115ms	115ms	115ms	115ms
Jitter	0ms	0ms	0ms	0ms	0ms

Table 4.83: 2Mb SSL VPN AES 256-bit SSH Latency and Jitter Results

4.8.3.3 Packet Overhead

Packet headers have remained constant across all SSL VPN tests, as can be seen in Table 4.84.

Attribute	Size (bytes)
IP Header	20
UDP Header	8
Security Layer	41
Avg. Packet Size	385.29

Table 4.84: 2Mb SSL VPN AES 256-bit SSH Packet Overhead Results

Table 4.84 shows an average packet size of 385.29 bytes, consistent with previous SSL VPN tests where all values were found to be in the range of 385-405 bytes. This would suggest that the average packet size causes no discernable performance degradation.

4.8.4 Conclusions

Table 4.85 shows the average performance metrics of the 2Mb SSL VPN AES 256-bit test, and indicates a reasonable performance level averaging at 62.26% – slightly higher than the 61.22% recorded for 2Mb SSL VPN AES 128-bit. When compared to its relative 8Mb test shown in Table 4.86, a noticeable performance difference amounting to around 15% can be seen – consistent with previous findings between all 8Mb and 2Mb tests.

Protocol	Throughput (KB/s)	Utilisation (%)
Video	161.13	62.94
Photo	148.41	57.97
Audio	161.54	63.10
Compressed	166.49	65.04

Table 4.85: 2Mb SSL VPN AES 256-bit Per File Type Summary Results

Now that all ciphers and protocols have been thoroughly tested, it is safe to conclude that the average performance drop of around 15% seen across all 2Mb results would support (Kuang, 2010) in his statement that increased latency does have a direct effect on throughput performance.

Protocol	**Throughput (KB/s)**	**Utilisation (%)**
Video	785.43	76.70
Photo	748.18	73.06
Audio	779.79	76.15
Compressed	809.07	79.01

Table 4.86: 8Mb SSL VPN AES 256-bit Per File Type Summary Results

Overall, like the AES 128-bit cipher, the AES 256-bit cipher is still able to achieve moderate performance across all protocols, with an average utilisation of 62.26%. Although this value is well outside the 70% level required to provide a reasonable performance level compared to PPTP, the higher grade SSL VPN ciphers provide much harder security. However, they incur much lower performance over low-bandwidth and high latency links, thus, are not suitable for small businesses that require high performance VPNs over such links.

Chapter 5

Summary and Conclusions

This section will present the final overall conclusions of the project based on the results collected during the experiment phase, as detailed in section 4. A brief summary of the project will be presented along with a final discussion on the findings of the experiment in relation to the research question and hypotheses. This section will conclude with the learned project limitations and any associated future works.

5.1 Brief Summary of Project

With Virtual Private Networks (VPNs) becoming an increasingly more popular asset used by organisations of all sizes, more information into their deployment and performance are required by IT professionals. Throughout this niche market there has been little experimentation or meaningful research into how different VPN tunnelling methods, or their ciphers, affect performance attributes – something which companies of all sizes require when producing a detailed case for VPN deployment in their organisation. However, it is small businesses that are missing out on the benefits of VPNs as they typically require a specialist with comprehensive knowledge of the subject to both setup and maintain such a configuration. Therefore a thorough investigation into popular and simple VPN tunnelling methods was required. This led to the following research question:

How do differing "traditional" PPTP and current SSL VPN tunnelling techniques affect the performance aspects of data communication based on modern small business network architecture?

To answer this question, an investigation into small business architectures and traffic patterns was conducted. This was followed by further investigations into the underlying technologies used in VPNs, as well as the performance aspects attributed to VPN tunnels. Upon review of the investigations, an experiment was conducted to test how the performance of different VPN tunnelling methods were affected by factors such as the level of *background* traffic, and different cryptographic ciphers. After conclusion of the experiment, the resulting data detailed in section 4 helped critically analyse how each VPN tunnelling method and its appropriate performance aspects – in relation to the research question, project objectives and hypotheses – affects the choices made not only small businesses, but organisations of all sizes.

5.2 Discussion of Results

5.2.1 Research Question Results

The results, available in section 4, indicated that PPTP was a very high performing VPN tunnelling method across low latency links running at 8Mbps. This is evident in section 4.1.4 where the average throughput across all protocols was recorded as 85%. However, the higher latency 2Mbps link resulted in a significant decrease in performance in throughput as detailed in section 4.2.4, where it only manages to average 68% utilisation of the maximum link speed. Latency, as previously discussed, was relatively low at a nominal 41ms with 0ms of jitter resulting in no discernable effect on the overall PPTP performance.

This would suggest that PPTP is an efficient and well performing VPN tunnelling method across low latency links only. However, PPTP has inherent security flaws as discussed in section 2.2.2.1, thus, cannot be considered a secure method of remote access into company networks. Nevertheless, due to its relative weak encryption cipher, it decreases the processing time needed to encrypt and decrypt every packet, resulting in the high performance on high bandwidth, low latency links.

When an organisation wishes to introduce VPNs, they are typically deployed for remote access into the company internal network. However, with PPTP and its security flaws, it cannot be considered cryptographically secure, thus, leaving your organisation vulnerable to attack. Therefore, it is not an appropriate choice if other, more secure, alternatives such as SSL VPNs are available.

The main advantage of SSL VPNs, such as OpenVPN, is its ability to implement multiple ciphers and varying degrees of key sizes for each cipher (generally, the larger the key size, the more secure the cipher). Blowfish is a 128-bit cipher and the default cipher in OpenVPN. Section 4.3.4 details the average values obtained when utilising the Blowfish cipher across a low latency 8Mbps link. It is able to average an utilisation value of around 84% – almost exactly the same as PPTP. Not only is an SSL VPN with the Blowfish cipher more secure, it is able to retain the same performance levels as PPTP.

When utilising the same SSL VPN setup over a higher latency 2Mbps link, incredibly, it is able to outperform the PPTP VPN by averaging an utilisation value of 70%. Therefore, it is reasonable to conclude that an SSL VPN is the most appropriate choice when deploying a business VPN. It is able to provide increased performance and security over the traditional PPTP method, whilst cutting down on the cost required to implement. This is due to OpenVPN being an open-source technology requiring no additional capital for licenses for every user connecting.

It can now be established that SSL VPNs can match (or surpass) the traditional PPTP tunnelling method in terms of speed and implementation cost when deployed using the Blowfish cipher. Nevertheless, some companies wish to transfer highly secretive and business critical information over the tunnels, such as the NSA (see section 2.2.2.2). Therefore, highly secure ciphers are required to provide the maximum level of security possible.

The performance of the AES 128-bit cipher over the lower latency 8Mbps link could be considered relatively good, with an average utilisation rate of 77.74% as detailed in section 4.4.4. This equates roughly to a performance drop of only 8% compared to PPTP and SSL with the

Blowfish cipher. With a higher key size of 256 bits, the AES 256-bit cipher averages only 70% utilisation on a low latency 8Mbps link – a noticeable further performance drop, now at 15% loss in performance over PPTP and SSL VPNs with the Blowfish cipher.

With the large savings incurred due to implementing open-source technologies, it could be argued that a 15% drop in performance is acceptable, with the knowledge that the VPN tunnel has the most secure cipher employed. Compounded with the ability to enable on-the-fly compression and the return to a less robust, but cryptographically secure cipher, it can be concluded that SSL VPNs provide the most appropriate combination of performance, security and cost for small businesses, as well as organisations of all sizes.

5.2.2 Hypotheses

The following hypotheses were discussed in section 1.2.6.

- SSL VPN will have reduced performance characteristics on all protocols compared to PPTP due to additional overheads.

Result: ***Confirmed***
Based on the results from the 8Mb PPTP and 8Mb SSL VPN Blowfish tests in sections 4.1.4 and 4.3.4 respectively, it is confirmed that all average throughput values for the SSL VPN were lower than PPTP. Blowfish is the best performing cipher for SSL VPNs; however, its average throughput rates across all protocols fell short of the PPTP results by 3%.

- Any performance drop on SSL VPN will be minimal (within a 15% margin) compared to PPTP.

Result: ***Confirmed***
Findings from section 4.5.4 which represent the 8Mb SSL VPN AES 256-bit results confirm this hypothesis with a recorded average utilisation value of 76.23% – only a 9% drop in performance compared to PPTP. The 2Mbps SSL VPN AES 256-bit results in section 4.8.4 display an average value of 62.26%, with the 2Mb PPTP results average reaching a value of 68% – well within the 15% margin specified in the hypothesis.

- SSL VPNs will be able to implement a wide array of robust encryption algorithms whilst still retaining an acceptable level of performance within the theorised 15% margin.

Result: ***Confirmed***
As stated in the above hypothesis, section 4.5.4 displays the most robust and secure SSL VPN running the AES 256-bit cipher which is able to obtain an average utilisation value of 76.23%, compared to 85% from PPTP, resulting in only a 9% drop in performance. Section 4.4.4 also displays the lower key size AES 128-bit variant achieving an average value of 77.74%, again, well within the theorised 15% margin based on PPTP results.

The same holds true for the 2Mb SSL VPN AES 256-bit test, where an average utilisation value of 62.26% is recorded. The equivalent 2Mb PPTP average utilisation stands at 68% – well within a 15% margin.

5.3 Project Limitations and Future Works

Although all of the authors hypotheses were found to be confirmed based on the experiment results detailed in section 4, the project was not without its limitations. One major limitation as discussed in section 3.2.2 was the inability to artificially inflate the latency without the reduction of link bandwidth. This was a limitation of the physical setup when using serial links to simulate an internet connection. To defeat this problem, a global test would need to be conducted to increase the latency (where distance directly correlates to increased latency). For example, from the author's location in Glasgow, the purchase of a server in Chicago would roughly equate to a latency of 110ms; however, due to project budget constraints and the inability to control the flow of data between endpoints, this scenario was not pursued. Nevertheless, stakeholders may wish to expand this study and implement this scenario to gain the most accurate results possible for high latency links.

Another important limitation stems from the inability to accurately simulate specific business-centric network traffic. In this experiment a pre-determined criteria of typical business network traffic was identified (see section 2.1.2); however, businesses have varying volumes of traffic and protocols in use on their network. Therefore, the results may not accurately reflect the results that all organisations would achieve. Nevertheless, the selected traffic patterns and protocols were subject to intense scrutiny during initial background investigations, and the selected traffic generation script accurately portrays the findings of that investigation as details in section 2.1.2.

5.4 Conclusion

This project was designed to provide a detailed analysis into the deployment of different VPN tunnelling techniques and report on their performance efficacies based on an average small business deployment. The critical analysis supplied throughout sections 4 and 5.2 offer comprehensive conclusions constructed from the research question and hypotheses. This was achieved with the help of a well-researched and articulate literature review and experiment methodology.

The results collected and detailed in section 4 would be of interest to organisations of all sizes. Although the study was conducted with small business scenarios, the differences between small and large scale VPN deployments are relatively minuscule, thus, the results are appropriate organisations of all sizes. Therefore, the performance aspects of VPNs are of importance to all businesses, and with many organisations in different fields employing VPN technologies, the results of this study will be of interest to a large range of organisations.

Furthermore, the findings of this report would also be of interest to individuals in Academia, as well as the public. VPNs are being widely deployed in Further Education providing staff with remote access to their network shares and documents. There has also been a large increase in private users utilising VPN technologies to provide security and anonymity over the internet. The use of VPNs in this sense allows users to surf the web anonymously, secure in the knowledge

that their identity and browsing history is not viewable by any third party.

What is evident from the undertaking of this study is that VPN usage will only increase as more organisations utilise the technology to reduce costs and increase efficiency. Through the execution of a logical and analytical project, a thorough and well-rounded critical analysis of the results and their relation to the research question and hypotheses was conducted in sections 4 and 5.2, providing the conclusion that SSL VPNs can be a cost-efficient and well-performing VPN alternative to traditional PPTP tunnelling methods.

Chapter 6

References

Anderson, D., 2000. Building Cisco Remote Access Networks, 1st ed. Indianapolis: Cisco Press.

Anon., 2004. VPN for the masses, Network Security, 2004(9), 3.

Barford, P., & Sommers, J., 2008. Packet-Loss Measurement.

Berger, T., 2006. Analysis of Current VPN Technologies, First International Conference on Availability, Reliability and Security.

Bingham, B., Strauss P., & Edwards, M., 2003. Validating the Business Benefits of Converged Communications.

Broderick J. S., 2001. Implementing Virtual Private Networks in Today's Organization, Information Security Technical Report, 6(1), pp.23-30.

Byron, K., 2008. Carrying too heavy a load? The communication and miscommunication of emotion by email. Academy of Management Review, 23, pp.309-27.

Casado, M. et al. 2007. Ethane: taking control of the enterprise, ACM SIGCOMM Computer Communication Review, 37(4).

Din, I. U., Mahfooz, S., & Adnan, M., 2009. Performance Evaluation of Different Ethernet LANs Connected by Switches and Hubs, European Journal of Scientific Research, 37(3), pp.461-70.

Evans, J., 2000. Performance evaluation of software virtual private networks (VPN), Proceedings 25th Annual IEEE Conference on Local Computer Networks, 2000, pp.522-23.

Feilner, M., 2006. OpenVPN: Building and Integrating Virtual Private Networks: Learn How to Build Secure VPNs Using this Powerful Open Source Application. Packt Publishing.

Fitzgerald, B., 2006. The transformation of open source software. MIS Quarterly, 30(3), pp.587-98.

Flood, J.E., 1998. Telecommunications Switching, Traffic and Networks, Chapter 4: Telecom-

munications Traffic. New York: Prentice-Hall, 1998.

Foley, M. J., 2010. "Behind the IDC data: Windows still No. 1 in server operating systems." 26 February 2010. Web. 11 March 2011. <http://www.zdnet.com/blog/microsoft/behind-the-idc-data-windows-still-no-1-in-server-operating-systems/5408>.

Forte, D., 2009. SSL VPN and return on investment: A possible combination, Network Security, 2009(10), pp.17-19.

Fraser, M., 2001. Understanding Virtual Private Networks (VPN).

Gilfeather, P. & Underwood, T., 2001. Fragmentation and High Performance IP. 15th International Parallel and Distributed Processing Symposium.

Gunkel, M. et al. 2008. Aggregation Networks: Cost Comparison of WDM Ring vs. Double Star Topology, Optical Network Design and Modeling.

Guo, X., & Zhai, Z., 2007. Investigation on Security of OpenVPN Architecture, Science Technology and Engineering.

Hall, M., 2008. Performance Analysis of OpenVPN on a Consumer Grade Router.

Hancock, B., 1997. Virtual private networks: What, why, when, where and how, Network Security, 1997(8), pp.8-11.

Hancock, B., 1999. ISPs and VPNs are at odds with each other, Computers & Security, 18(5), pp.376-80.

Harding, A., 2003. SSL Virtual Private Networks, Computers & Security, 22(5), pp.416-20.

Harmening, J. & Wright, J., 2009. Computer and Information Security Handbook, Bosting: Morgan Kaufmann.

Hunt, R., 2001. Technological infrastructure for PKI and digital certification, Computer Communications, 24(14), pp.1460-71.

Jain, M. & Dovrolis. C., 2002. End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput, Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications, 2002.

Keng Lim, L. et al. 2001. Customizable virtual private network service with QoS, Computer Networks, 36(2), pp.137-51.

Khanvilkar, S. & Khokhar, A., 2004. Virtual private networks: an overview with performance evaluation, Communications Magazine, 42(10), pp.146-54.

Khare, R., 1998. I want my FTP: bits on demand, Internet Computing, 2(4), pp.88-91.

Kuang, J., & Bhuyan, L., 2010. LATA: a latency and throughput-aware packet processing system, Proceedings of the 47th Design Automation Conference.

Liu, J. et al. 2009. A real-time network simulation infrastructure based on OpenVPN, Journal of Systems and Software, 82(3), pp.473-85.

Lu, C., & Tseng, S., 2002. Integrated Design of AES (Advanced Encryption Standard) Encrypter and Decrypter, 13th IEEE International Conference on Application-Specific Systems, Architectures and Processors.

Milanovic, S., 2001. Deploying IP-based Virtual Private Network across the Global Corporation. Communications World.

Munro, K., 2006. VPN security needs beefing up, Infosecurity Today, 3(4), 40.

Olson, P., 2011. Google Gobbles Internet Explorer's Market Share With Chrome, Forbes.

Poon, S., 2008. Future of Small Business E-Commerce. Electronic Commerce: Concepts, Methodologies, Tools, and Applications, pp.1.

Rowan, T., 2007. VPN technology: IPSEC vs SSL, 2007(12), pp.13-7.

Schneier, B., 1998. Cryptanalysis of Microsoft's point-to-point tunneling protocol (PPTP). Proceedings of the 5th ACM conference on Computer and communications security.

Schneier, B., Mudge, &d Wagner, D., 1999. Cryptanalysis of Microsoft's PPTP Authentication Extensions (MS-CHAPv2), Computer Science, 1740.

Strayer, T., 2004. Privacy issues in virtual private networks, Computer Communications, 27(6), pp.517-21.

Truchan, L. C, 1993. Experimental Design and Testing: Hatching and Development in Brine Shrimp.

Tyson, J., 2004. How internet infrastructure works.

Uhlig, S., & Bonaventure, O., 2000. On the Cost of Using MPLS for Interdomain Trafic, Computer Science, 1922, pp.141-52.

Utter, JC., & Snyder, J., 2003. Remote Access Services (RAS) vs Network Infrastructure.

Wright, M., 2000. Virtual Private Network Security, Network Security, 2000(7), pp.11-14.

Yonan, J., 2004. 'Overhead added to each packet by OpenVPN?" 30 November 2004. Web. 12 March 2011. <http://openvpn.net/archive/openvpn-users/2004-11/msg00649.html>.

Chapter 7

Bibliography

Cheung, K. H. & Misic, J., 2002. On virtual private networks security design issues, Computer Networks, 38(2), pp.165-79.

Cui, W. & Bassiouni, M., 2003. Virtual private network bandwidth management with traffic prediction, Computer Networks, 42(6), pp.765-78.

Donovan, S., Drabwell, P. & Harbird, R., 2001. VPNs and Lightweight Clients, Information Security Technical Report, 6(1), pp.49-64.

Henmi, A. et al. 2006. Firewall Policies and VPN Configurations. Burlington: Syngress.

Hucaby, D., 2007. CCNP BCMSN Official Exam Certification Guide, 3rd ed. Indianapolis: Cisco Press.

Knight, W., 2005. VPN boom bewilders users, Infosecurity Today, 2(3), pp.15-19.

Knipp, E. et al. 2002. Managing Cisco Network Security, 2nd ed. Burlington: Syngress.

Morgan, B. & Lovering, N., 2008. CCNP ISCW Official Exam Certification Guide, 3rd ed. Indianapolis: Cisco Press.

Qiang, H., Frahim, J. & Waheed, A., 2008. SSL Remote Access VPNs, 1st ed. Indianapolis: Cisco Press.

Ranjbar, A., 2007. CCNP ONT Official Exam Certification Guide, 4th ed. Indianapolis: Cisco Press.

Rescorla, E., 2001. SSL and TLS: designing and building secure systems.

Rittinghouse, J. & Hancock, W., 2004. Cybersecurity Operations Handbook. Burlington: Digital Press.

Rowan, T., 2007. VPN technology: IPSEC vs SSL, Network Security, 2007(12), pp.13-17.

Ruest, D. & Ruest, N., 2008. Microsoft Windows server 2008: the complete reference.

Snader, J., 2005. VPNs Illustrated: Tunnels, VPNs, and IPsec, 1st ed. Addison Wesley.

Stewart, B. & Gough, C., 2008. CCNP BSCI Official Exam Certification Guide, 4th ed. Indianapolis: Cisco Press.

Viega J., Messier, M., & Chandra, P., 2002. Network security with OpenSSL, 1st ed. Sebastopol: O'Reilly Media.

Waite, S., 2006. Securing online business with SSL, Network Security, 2006(3), pp.10-12.

Watkins, M. & Wallace, K., 2008. CCNA Security Official Exam Certification Guide, 5th ed. Indianapolis: Cisco Press.

Wendell, O., Healy, R. & Donohue, D., 2010. CCIE Routing and Switching Certification Guide, 4th ed. Indianapolis: Cisco Press.

Appendix A

Appendices

A.1 Router 1 - 8Mbps

```
!
version 12.4
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Router1
!
interface FastEthernet0/0
 no ip address
 duplex auto
 speed auto
 shutdown
!
interface FastEthernet0/1
 ip address 172.16.0.1 255.255.255.0
 duplex auto
 speed auto
!
interface Serial0/2/0
 ip address 10.0.0.1 255.255.255.252
 clock rate 8000000
!
interface Serial0/2/1
 no ip address
 shutdown
!
interface Vlan1
 no ip address
 shutdown
!
ip classless
```

```
!
line con 0
line vty 0 4
 login
!
end
```

A.2 Router 2 - 8Mbps

```
!
version 12.4
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Router2
!
interface FastEthernet0/0
 no ip address
 duplex auto
 speed auto
 shutdown
!
interface FastEthernet0/1
 no ip address
 duplex auto
 speed auto
 shutdown
!
interface Serial0/2/0
 ip address 10.0.0.2 255.255.255.252
!
interface Serial0/2/1
 no ip address
 shutdown
!
interface Vlan1
 no ip address
 shutdown
!
ip classless
!
line con 0
line vty 0 4
 login
!
end
```

A.3 Traffic Generation Config - 8Mbps

```
fastethernet0/0
add tcp
rate 1000
datalink ios-dependent fastethernet0/0.10
12-arp-for 172.16.0.2
13-src 172.16.0.1
13-dest 172.16.0.2
length random 16 to 1500
burst on
burst duration off 1000 to 2000
burst duration on 1000 to 3000
add fastethernet0/0 1
14-dest 80
data ascii 0 GET /index.html HTTP/1.1
add fastethernet0/0 1
14-dest 21
add fastethernet0/0 1
14-dest 110
add fastethernet0/0 1
14-dest 25
add fastethernet0/0 1
14-dest 53
```

A.4 Switch Config

```
!
version 12.2
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Switch
!
interface FastEthernet0/1
!
interface FastEthernet0/2
!
interface FastEthernet0/3
!
interface FastEthernet0/4
!
interface FastEthernet0/5
!
interface FastEthernet0/6
!
interface FastEthernet0/7
```

```
!
interface FastEthernet0/8
!
interface FastEthernet0/9
!
interface FastEthernet0/10
!
interface FastEthernet0/11
!
interface FastEthernet0/12
!
interface FastEthernet0/13
!
interface FastEthernet0/14
!
interface FastEthernet0/15
!
interface FastEthernet0/16
!
interface FastEthernet0/17
!
interface FastEthernet0/18
!
interface FastEthernet0/19
!
interface FastEthernet0/20
!
interface FastEthernet0/21
!
interface FastEthernet0/22
!
interface FastEthernet0/23
!
interface FastEthernet0/24
!
interface GigabitEthernet1/1
!
interface GigabitEthernet1/2
!
interface Vlan1
 no ip address
 shutdown
!
line con 0
!
line vty 0 4
 login
line vty 5 15
 login
!
end
```

A.5 Router 1 Config - 2Mbps

```
!
version 12.4
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Router1
!
interface FastEthernet0/0
 no ip address
 duplex auto
 speed auto
 shutdown
!
interface FastEthernet0/1
 ip address 172.16.0.1 255.255.255.0
 duplex auto
 speed auto
!
interface Serial0/2/0
 ip address 10.0.0.1 255.255.255.252
 clock rate 2000000
!
interface Serial0/2/1
 no ip address
 shutdown
!
interface Vlan1
 no ip address
 shutdown
!
ip classless
!
line con 0
line vty 0 4
 login
!
end
```

A.6 Router 2 Config - 2Mbps

```
!
version 12.4
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
```

```
!
hostname Router2
!
interface FastEthernet0/0
 no ip address
 duplex auto
 speed auto
 shutdown
!
interface FastEthernet0/1
 no ip address
 duplex auto
 speed auto
 shutdown
!
interface Serial0/2/0
 ip address 10.0.0.2 255.255.255.252
!
interface Serial0/2/1
 no ip address
 shutdown
!
interface Vlan1
 no ip address
 shutdown
!
ip classless
!
line con 0
line vty 0 4
 login
!
end
```

A.7 Traffic Generation Config - 2Mbps

```
fastethernet0/0
add tcp
rate 200
datalink ios-dependent fastethernet0/0.10
12-arp-for 172.16.0.2
13-src 172.16.0.1
13-dest 172.16.0.2
length random 16 to 1500
burst on
burst duration off 1000 to 2000
burst duration on 1000 to 3000
add fastethernet0/0 1
14-dest 80
```

```
data ascii 0 GET /index.html HTTP/1.1
add fastethernet0/0 1
14-dest 21
add fastethernet0/0 1
14-dest 110
add fastethernet0/0 1
14-dest 25
add fastethernet0/0 1
14-dest 53
```

A.8 OpenVPN Client Config

```
client
pull
proto udp
dev tun0
ca ca.crt
cert server.crt
key server.key
dh dh2048.pem

ping 5
ping-restart 45
ping-timer-rem

## 3 CIPHERS ##
cipher BF-CBC
#cipher AES-128-CBC
#cipher AES-256-CBC
comp-lzo
```

A.9 OpenVPN Server Config

```
mode server
proto udp
local 172.16.0.2
port 1194
dev tun0
ca ca.crt
cert server.crt
key server.key
dh dh2048.pem

topology subnet
server 10.10.10.0 255.255.255.0
```

```
push "redirect-gateway def1"
push "route-gateway 10.8.0.1"
push "dhcp-option DNS 10.8.0.1"
push "dhcp-option DOMAIN honours.project.com"
push "topology subnet"
push "ping 5"
push "ping-restart 30"
push "ping-timer-rem"

ping 5
ping-restart 45
ping-timer-rem

## 3 CIPHERS ##
cipher BF-CBC
#cipher AES-128-CBC
#cipher AES-256-CBC
comp-lzo
```