Ekaterina Danilova

Comparing the performance of different VPN technologies with TLS/SSL

Bachelor’s Thesis
INFORMATION TECHNOLOGY

November 2015

MAMK
University of Applied Sciences

DESCRIPTION

Date of the bachelor's thesis

27 November 2015

Author(s)

Ekaterina Danilova

Degree programme and option

Information Technology

Name of the bachelor's thesis

Comparing the performance of different VPN technologies with TLS/SSL

Abstract

The idea of the thesis was investigating various types of popular VPN technologies and products. The goal was to find out the most comfortable, high-performing and easy-to-install solution. In the end of the thesis I made conclusion on the products I investigated and gave suggestions about the suitable conditions for the use these products. In order to find out the performance and usability if the products I held various tests.

Subject headings, (keywords)

Network Security, SSL, TLS, VPN

Pages 44	**Language** English	**URN**

Remarks, notes on appendices

Tutor Matti Koivisto	**Employer of the bachelor's thesis**

Contents

1 INTRODUCTION

As more and more companies are using the Internet for their operations, the network security becomes a very important topic. There are various ways to achieve network security, such as anti-virus software, firewalls, encrypting the traffic and many others. Traffic encryption is one of the extremely important components of network security of an organization, because no other tools will protect the data if it can be listened to in unencrypted form. Nowadays one of the best solutions for encrypting traffic is the VPN technology.

VPNs became very popular for various reasons. The first one, and the very reason why the idea of VPNs appeared in the first place, is the need to have remote access to different networks. For example, many companies have remote workers who might need to access the company's network resources without being directly connected to the company's network. Clientless VPN solutions (such as OpenVPN) perfectly serve this purpose of remotely and securely connecting to other networks.

Also, VPNs are necessary when there is need to connect different LANs through the public Internet. The most typical example is big companies that have offices all over the world and all those offices require access to the same network resources. Leased lines and other solutions are extremely expensive for this goal and impossible for smaller businesses, and therefore VPN is used for connecting networks through the public Internet.

And the main perspective from which I will look at VPNs in this thesis comes from the third need — the need for secure communications. VPN encrypts all the traffic which is sent, and that is why it is nowadays used very often for security purposes. Not only big companies from the two above mentioned examples use this advantage of VPN, but also normal people use it to get more secure communications.

These are the main reasons why VPN technologies are so popular and widely used nowadays and why this topic is essential in the fields of networking and network security. Even though different VPN solutions serve the same purpose, their working principles differ a lot, which makes it important to understand the various ways of establishing VPN connections and the advantages and disadvantages of the different VPN options available (Scott et al. 1999).

1.1 Aim of the study

The theoretical aim consists of learning about various VPN types, their differences and technologies. The attention will be mainly put on the SSL/TLS technology, its working principle, advantages and disadvantages. However, other VPN technologies will be studied as well. The aim is to describe advantages and disadvantages of different VPN technologies and to describe the SSL/TLS in comparison to them.

The practical aim includes testing given VPN technology performance on different devices and in different conditions. Also different types of VPN solutions will be tested as well, such as Cisco VPN and OpenVPN. The performance of mobile devices and laptops differs a lot and is very important, and therefore the VPNs will be tested on various kinds of such devices in order to find solutions which provide the highest efficiency.

1.2 Structure of the study

My thesis starts with an introduction which deals with today's IT field and the place of VPN technologies in it. Then comes Chapter 2 where I take a closer look at VPN, its history, applications and usage. Also, it contains the comparison of different VPN types and the ways of VPN classifications. I am going to compare various VPN technologies, their working principle, advantages and disadvantages. Chapter 3 concentrates on specific VPN technologies, SSL/TLS, and describes them in detail. In Chapter 4 I will continue the topic of SSL VPN and take a look at the security of this technology, possible attacks and weaknesses. Chapter 5 focuses on describing the specific SSL VPN technologies that will be used in measurements.

The practical part starts from Chapter 6 which includes the set up of the technologies mentioned in the previous chapter with instructions and the description of the process as well as the measurement results. Chapter 7 will follow exactly the same plan, but it will focus on mobile devices. In Chapter 8 I will compare the measurement results of the two previous chapters and make conclusions based on this comparison in Chapter 9.

2 VPN SOLUTIONS

VPN is not just a single technology, it is a combination of many different types that use different technologies. But before I start to describe VPN types it is important to understand what the idea of VPN is in general.

2.1 What is VPN?

According to VPN Consortium (2008) VPN is defined as “a private data network that makes use of the public telecommunication infrastructure, maintaining privacy through the use of a tunneling protocol and security procedures”. In other words, VPN connects networks using the public Internet infrastructure. In order to achieve the functionality of leased lines VPN uses traffic encryption and different tunneling protocols. VPN technology can be used for different purposes, and now as it has spread, it has become easy to use not only for big companies who need to connect branch offices or remote workers, but also for ordinary people who are worried about their security. The main reason for that is that VPN provides very hight level of privacy protection for a reasonable price and quite low technical knowledge requirements. In addition to this, even ISPs nowadays provide paid VPN services, and therefore no specific technical knowledge is required for this technology.

Mobile companies were the first sphere where the idea of a technology similar to VPN was invented. Those phone companies provided privately shared resources for voice messages. VPN uses the same idea, but instead of voice messages this idea is applied to data. (Ferguson & Huston 1998) .

The need for VPNs appeared as soon as companies started to use networks in their operations. The main predecessor of VPN was leased lines technology. It was not popular because of high costs, and only big companies could afford it. Also, it was not very flexible and as soon as a leased line was installed all the changes in location would be very expensive and require moving this line. However, leased lines are not dead and are used by banks and very big companies who require very high bandwidth and cannot risk using the public Internet for operations either because of security or because of the risks of bandwidth drop etc. Therefore, even though leased line technology is not widely used nowadays it still exists, but in the small/middle-sized businesses and in the private sector it was replaced by VPNs. (Ferguson et al. 1998, 4.)

2.2 VPN classification

VPNs are classified in a number of different ways, based on their technology, security levels or deployment type. For example, Günter Manuel (1998) groups them according to the purpose as well as the implementation and the size of a company as shown in Table 1:

Purpose:	Remote access network
	Branch office connection network
	Business partner/supplier networks
Implementation and size of the company:	VPNs offered by ISP
	VPNs implemented by company itself

TABLE 1. VPN Classification (Manuel 1998)

The above classifications used to be popular before, but now many more ways of classifying VPNs have appeared as shown in Table 2:

OSI layers:	Layer 4/7 VPN – WebVPN
	Layer 3 VPN – IPSec
	Layer 2 VPN – L2TP, PPTP, MPPE
Compulsory tunnel mode/voluntary tunnel mode:	Compulsory mode – L2F, PPTP, L2TPv2/L2TPv3
	Voluntary mode – SSL/TLS, IPSec, L2TPv2/L2TPv3, PPTP

TABLE 2. Alternative VPN Classifications (Lewis 2006)

Some of these classifications are a bit old-fashioned and are not widely used nowadays, for example, the purpose classification which was defined by Günter Manuel. Some other classifications are more widely used, for example classifications based on OSI layers. When choosing a VPN solution different methods of classifying should be taken into account in order to get a fuller picture of the advantages and disadvantages of selected method.

2.3 Overview of the VPN protocols

VPN uses a variety of protocols which are used for tunnelling or security purposes. The tunnelling and security protocols are:

- PPTP (Point-to-point Tunnelling Protocol)
- L2TP (Layer 2 Tunnelling Protocol)
- IPSec (IP Security)
- SSL/TLS (Secure Sockets Layer/Transport Layer Security)

Some protocols are more popular and some are less, and the list includes the main technologies that are in use. Nowadays IPSec is a popular solution, SSL/TLS's popularity is growing, PPTP is still widely used despite serious security problems and L2TP is quite popular, too. As all these protocols are used, all of them are worth mentioning and explaining.

2.4 PPTP

PPTP is a popular protocol developed by Microsoft as an extension to PPP which has been in use for many years already. PPTP is not a protocol suite, but just a single protocol, and therefore it is used together with other protocols to provide security. For example, the most popular authentication protocol which is used with PPTP is MS-CHAP v.2.

Moving on to describe the architecture of PPTP under Windows Server, the PPTP uses the following steps to set up secure communication:
PPTP Connection and Communication
PPTP uses PPP which is a remote access protocol to set the connection and to encrypt packages.

PPTP Control Connection
PPTP tunnel is set between the PPTP server and the client using the PPP connection established in the previous step. PPTP

PPTP Data Tunnelling
IP datagrams with encrypted PPP packets are exchanged through the PPTP tunnel.

PPTP was available as a standard VPN solution for Windows without the need for additional software and was very easy to set up, which helped it to gain great popularity. However, PPTP has very weak security and most of its vulnerabilities come from MS-CHAP. MS-CHAP v1 was deprecated in Windows Vista (Microsoft 2007) due to its insecurity and MS-CHAPv2 contains serious vulnerabilities (Microsoft 2012). In short, this protocol is still used because it is fast, easy to set up and has no compatibility problems usually. However, PPTP is a very bad option and should never be selected, because the authentication protocol that it uses is not safe. A lot of weaknesses of this protocol were explained by Kevin Townsend in his security analysis of this protocol (Townsend 1998).

2.5 L2TP and L2TP/IPSec

L2TP is the tunneling protocol of the 2nd OSI layer which does not provide data encryption, and usually it is used together with the encryption protocol IPSec. Historically L2TP was developed from two protocols: PPTP (Point-to-point Tunnelling Protocol) and L2F (Layer 2 Forwarding Protocol). This is the reason why the overall working principle of all the protocols mentioned is practically the same except for some details. However, because these details are inside the mechanisms of the protocol, it is impossible to describe it without through explanation of the protocols stated above. (Shea 2000, 9.)

As L2TP alone cannot provide security, as stated in RFC 3193, it must be used only together with IPSec which has secure encryption mechanisms (Patel 2001). IPSec does not have huge weaknesses, and therefore this solution is safe and also easy to install. However, the packets are encapsulated twice which slows down this protocol and makes it less efficient than other solutions.

2.6 IPSec

Even though IPSec is often used in pair with L2TP, it can be used as a tunnelling protocol by itself. IPSec is a framework for a set of protocols that operates on the 3rd layer of the OSI model. It can work in either the Transport or Tunnel modes. IPSec achieves safe data transfer by using cryptographic keys in the beginning of the tunnelling session and encrypting data with these keys.

It has a big advantage of being transparent to the user, e.g. it is not required to make any connections like in SSL in the browser.
IPSec uses several protocols such as:

- AH (Authentication Headers) which identifies that data arrived unchanged and from the right source.

- ESP (Encapsulating Security Payloads) which encrypts data in order to protect it from eavesdropping.

- ISAKMP (Internet Security Association and Key Management Protocol) which provides a secure method of key exchange.

IPSec is a very popular choice nowadays because of its high level of security and easy maintenance. However, it has some disadvantages like requirements for higher processing power and rare compatibility issues. (Frankel 2005, 4-15.)

2.7 SSL

SSL technology is becoming more and more popular. One of the main reasons for this is that it does not require additional client software installed. Clientless SSL does not require any software installed except for the web-browser, which makes it easy to use and a very good option, for example, for remote workers. However, SSL VPN is more limited in some sense. Unlike in IPSec, for example, it does not give the access to the whole network, but just for web-based applications. If non-browser applications require the use of VPN, additional plug-ins should be installed. This point can be seen both as an advantage and disadvantage: on one hand, security is improved because the client is more restricted in access. On the other hand, installing additional software might be required.

Nowadays SSL is mostly used in the following situations: for access to websites which require user authentication, for safe connection to e-commerce websites, for safe access to email or databases and for remote access.

3 OVERVIEW OF THE SSL/TLS TECHNOLOGY

The main objective of my thesis lies in the area of the SSL/TLS technologies. Therefore, describing those technologies is essential for the understanding of my thesis topic.

3.1 What are the TLS and SSL protocols?

TLS and SSL are cryptographic protocols that are used for creating secure VPN tunnels by providing data encryption and authentication. SSL was the first one to appear and it was developed by Netscape. SSL v.2 was developed in February 1995 and proved to be very insecure and was forbidden by RFC 6176 (Turner & Polk 2011). The latest version of SSL is 3.0 and released by Netscape in 1996, but as Netscape does not exist any more, new SSL specifications will not be developed. Therefore, TLS v.1.0 was created by IETF in 1999 as a “SSL 3.1”. It is more secure, but still it did not replace SSL completely. The latest version nowadays is TLS v. 1.2 and it was released in August 2008.

3.2 SSL

According to SANS Institute (2003) SSL is defined as a “secure communications protocol of choice for a large part of the Internet community. There are many applications of SSL in existence, since it is capable of securing any transmission over TCP. Secure HTTP, or HTTPS, is a familiar application of SSL in e-commerce or password transactions.” Nowadays SSL is one of the most popular technologies and used in many wide-spread different VPN solutions: from IOS VPN to OpenVPN. Both of them will be discussed later in the thesis. (Sans Institute 2003, 3.)

In order to provide security SSL uses different techniques: encryption for keeping privacy, the use of certificates for identity authentication and message integrity checking to provide higher reliability. Understanding the SSL working principle demands knowing and separating two different terms: SSL connection and SSL session. The SSL session is a set of negotiated cryptography parameters between the server and client. The use of sessions makes it possible to avoid setting parameters for every connection. Connection is associated with only one session and refers to the live communication channel between a server and a client. Session lasts longer than a connection and can contain a lot of them. These terms will be used later when explaining SSL functionality.

SSL is not just a single protocol, but several layers of them. Table 3 describes SSL protocol architecture:

<table>
<tr><td>SSL Handshake Protocol</td><td>SSL Change Cipher Spec Protocol</td><td>SSL Alert Protocol</td></tr>
<tr><td colspan="3">SSL Record Protocol</td></tr>
<tr><td colspan="3">TCP</td></tr>
<tr><td colspan="3">IP</td></tr>
</table>

TABLE 3. SSL Protocol architecture (Microsoft 2003)

Table 3 shows that SSL has two layers of protocols. The lower one is SSL Record Protocol that contains basic security methods which are used by the higher level protocols of SSL. The higher level one is Handshake Protocol which has three sub-protocols as follows:

Handshake Protocol is used to exchange session information between the server and client. It sends the session ID, certificates, cipher specification, compression algorithm and shared secret.

Change Cipher Spec Protocol is used to change the parameters of connection. It generates a 1-bit message with the value of 1 and leads to updating encryption used for the connection.

Alert Protocol is used for alert messages. It generates a 2-bit message which contains the error code. If the error code shows an error that cannot be fixed, the current SSL session is destroyed.

The SSL Record protocol is the most important for understanding the SSL working principle. It is used to provide authentication, agree on encryption methods and to calculate the master key. This protocol is used before the applications are able to exchange data. As this thesis focuses on the SSL/TLS protocols it is important to understand the working principle and its handshake process, which is defined in the Oracle documentation (2010) as follows:

1. Client Hello

2. Server Hello
3. Authentication and Pre-master Secret:
    1. *Does today belong to the validity period?*
    2. *Is CA (Certificate Authority) trusted?*
    3. *Does the CA's public key validate the issuer's digital signature?*
    4. *Do domain names of the server and the server's certificate match?*
    5. *The server is authenticated*
4. Decryption and Master Secret
5. Generate Session keys
6. Encryption with Session keys

I will describe the handshake more closely and look into each step of this process.

Client hello

The client sends information to the server. The information includes: client’s SSL version, cipher settings, randomly generated data and other information that the server needs to communicate via SSL.

Server hello

The server sends its SSL version number, cipher settings, randomly generated data, additional information that the client needs to communicate via SSL. Also, the server sends its certificate. If the client wants to access server resource that needs the client's authentication, the server sends the request for the client's certificate.

Authentication and Pre-master Secret

Authentication has several steps and consists of several tests which must be passed successfully:

Does today belong to the validity period?

The date is checked and the phase is passed successfully if it belongs to the validity period of the certificate.

Is CA (Certificate Authority) trusted?

Every SSL-enabled client keeps the list of trusted CAs. If the DN (Distinguished Name) of issuing certificate matches the DN of the CA from the trusted list, the CA is considered to be trusted.

Does the CA's public key validate the issuer's digital signature?

During this step the client checks several things: if the CA's certificate's public key corresponds to the CA's private key that was used to sign the server certificate and if the information in the server certificate has not been changed since the certificate was signed by CA. If at least one of these checks is not passed, the authentication fails.

Do domain names of the server and the server's certificate match?

This step provides protection against Man-in-the-Middle attacks because during this step it is checked if the server's certificate points to the network address where the server is located. In case the domain names do not match the authentication cannot be passed.

The server is authenticated

If all the above requirements are met, the server is authenticated and the handshake process continues. If at least one of the requirements was not met, the user is informed about the failure to establish secure connection. Also, there is possibility that the server requires additional clients' authentication, which means that additional authentication process might be required. However, I do not describe this process, because it is not a compulsory part of the Handshake process.

The above process of server's authentication is described in Figure 1 below:

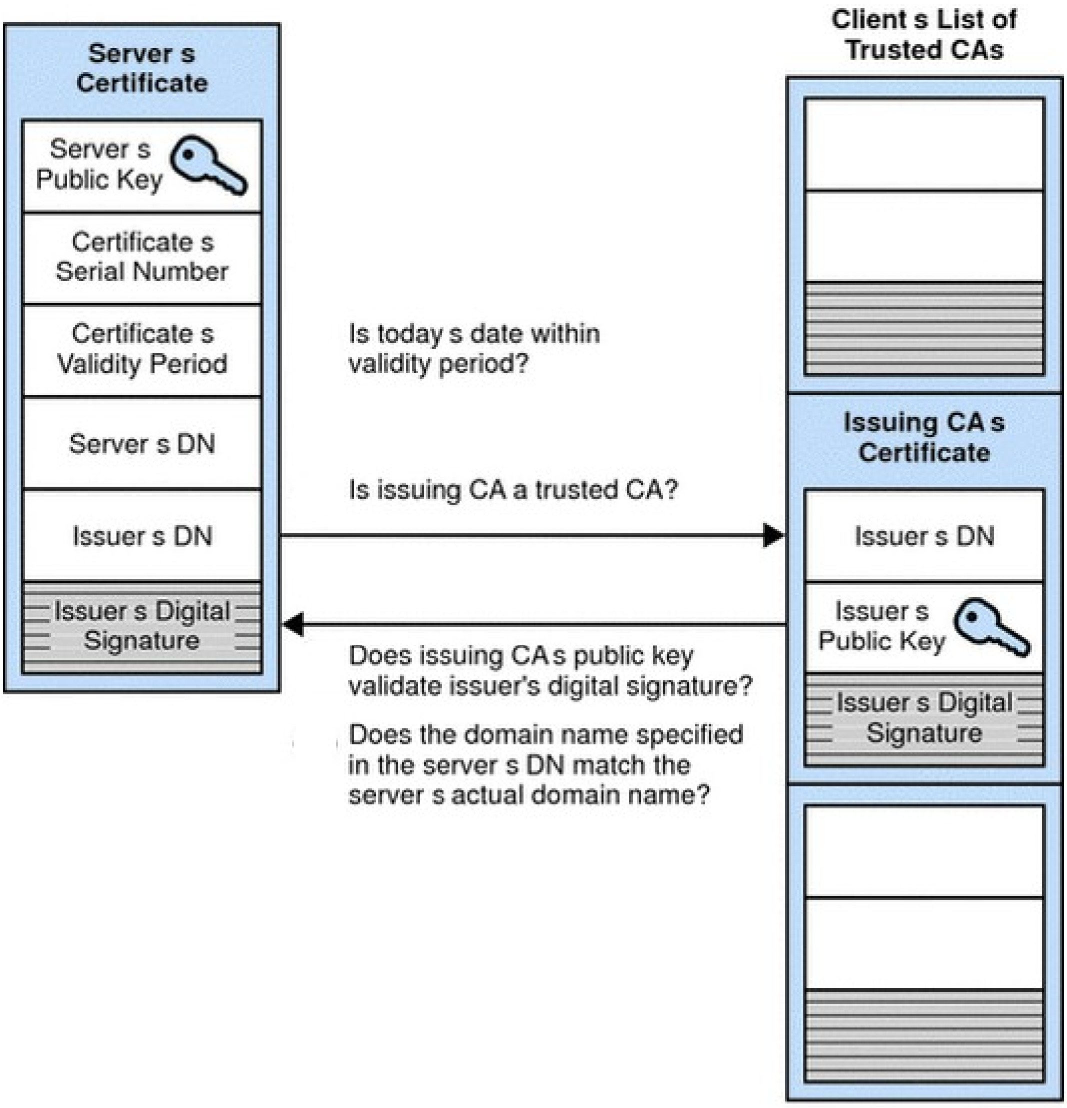

Server s Certificate
Server s Public Key
Certificate s Serial Number
Certificate s Validity Period
Server s DN
Issuer s DN
Issuer s Digital Signature
Client s List of Trusted CAs
Is today s date within validity period?
Is issuing CA a trusted CA?
Does issuing CAs public key validate issuer's digital signature?
Does the domain name specified in the server s DN match the server s actual domain name?
Issuing CA s Certificate
Issuer s DN
Issuer s Public Key
Issuer s Digital Signature

FIGURE 1. Server authentication during the SSL Handshake (Microsoft Corporation 2003)

If the authentication is passed successfully, the client with the cooperation of the server creates Pre-master secret for the session based on the cipher used. Then Pre-master Secret is encrypted with the server's public key which was received by the server in the Server's hello step and after this it is sent to the server.

The server might also require client authentication, which is not a compulsory step of the SSL handshake. In this case the client signs one more piece of data which is sent together with the client's certificate to the server.

Decryption and Master Secret

The server decrypts Pre-master Secret using its private key in order to create the master secret.

Generate Session keys

Session keys are symmetric keys which are used for encryption, decryption and verifying the integrity of data that is transferred during the SSL session.

Encryption with Session key

The server and client inform each other that Session keys were successfully created. Then the server and client exchange messages encrypted with these keys.

From now the SSL session has started and Session key will be used for encryption and integrity checks.

3.3 TLS

TLS was developed by IETF (Internet Engineering Task Force) and basically it is an upgrade to SSL v.3 (Thomas 2000). TLS does not have big differences compared to the SSL technology, but it has a lot of improvements. The biggest differences are:

- New Alert messages have been added.
- While SSL uses MD5 and SHA, TLS uses H-MAC.
- Keys are generated differently: SSL uses RSA, Diffie-Hellman or Fortrezza and TLS uses the HMAC standard and the pseudo random function.
- TLS has an easy way of passing the certificate verification message, while SSL requires a more complicated procedure for that.
- Some messages have different fields.

Except for these differences, TLS has exactly the same working principle as SSL, and therefore I will not describe it here, and later the term SSL will stand for both SSL and TLS.

4 SSL/TLS SECURITY

As SSL is a very popular VPN technology nowadays, it is attacked a lot. Even though this technology is considered to be secure, different weaknesses are discovered and fixed from time to time. There are many possible attacks to the SSL protocol which will be described in this chapter.

The up-to-date list of TLS attacks is summarized in RFC 7457 (Holz & Sheffer & Saint-Andre 2015). Some of the weaknesses are shared with SSL thanks to the similarities of the technologies. One such example is attacks on the RC4 algorithm which was shared by both technologies and remained as an issue for a long time. The following section describes the most widely-spread and dangerous types of possible attacks on TCP/IP.

4.1 Attacks on RC4

RC4 is a popular cipher suite which has been long known for its weakness, but it is still used in TLS, SSL and WEP. One of the most famous attacks on it took place in 2001 and proved that the key can be broken after the long analysis of traffic (Fluhrer & Mantin & Shamir 2001). The problem is caused by the fact that the bytes in traffic are not random enough and using the same message many times makes it possible to find enough patterns to decrypt it (Mironov 2002). However, this attack requires a big amount of traffic and can be avoided by certain measures such as using the RC4-drop algorithm, which is exactly the same as RC4, except that the first “unsafe” bytes are discarded after key scheduling.

4.2 SSL Stripping

The basic idea of SSL Stripping lies in downgrading the website from HTTPS to HTTP, this way preventing the user from using a protected SSL connection. This attack is the type of Man-in-the-Middle attacks, because it includes the attacker interfering in the connection and routing the victim's traffic through his own created proxy server. The scheme of the attacker rerouting the traffic is available below:

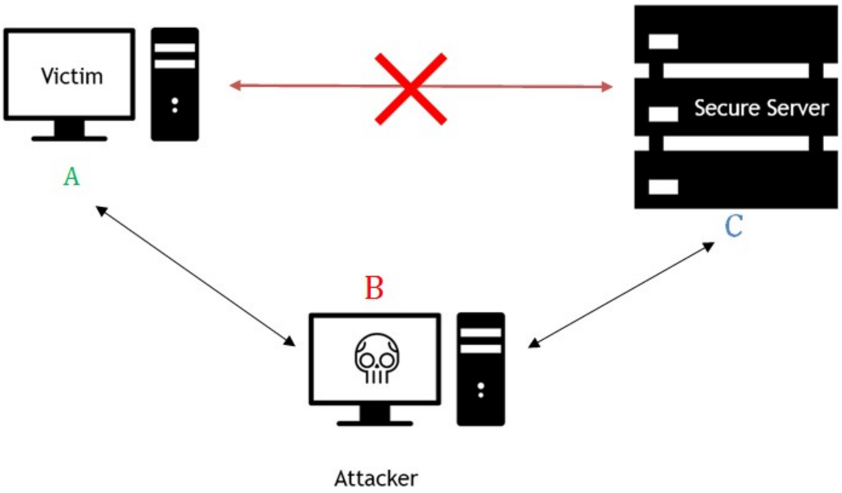


FIGURE 2 SSL stripping (Avinash 2015)

The basic working principle is the attacker in the middle who transfers the messages between the victim and the server and changes the HTTPS connection to a plain-text HTTP in the process. First, the victim tries to access the server (netbank, for example) and his request comes to the attacker (B) who forwards it to the server. Then the server sends the reply which is supposed to forward the victim to protected HTTPS page, but it doesn't happen, because the attacker replaces HTTPS with HTTP in the server's reply and forwards it to the victim. Since the victim is now using HTTP requests, they are not protected anymore and the attacker can steal valuable data. This attack was first described in 2009 by Moxie Markinspike in the Black Hat conference in 2009. (Avinash 2015).

4.3 Implementation attacks

A lot of attacks happen, because the protocol is not implemented properly. That is why implementation attacks are worth mentioning. For example, the latest loud attack to SSL – Heartbleed - is a type of implementation attack. The problem of Heartbleed is caused by fixed buffer size, which can show extra unnecessary information if the buffer is not completely filled with the reply, which can give the attacker sensitive information. As the issue was caused by a coding mistake, it belongs to the group of implementation attacks.

A big number of implementation attacks is caused by various misunderstandings of SSL API, which leads to most of the mistakes. The big list of popular services which have the implementation errors is listed in the study “The most dangerous code in the world: validating SSL certificates in non-browser software" (Georgiev 2012).

4.4 Compression attacks

There are several attacks which are related to HTTPS compresson: CRIME, TIME and BREACH. The first one stands for Compression Ratio Info-leak Made Easy and is caused by the compression of cookies when using the HTTPS and SPDY (developed by Google to manipulate HTTP traffic in order to improve security and speed) protocols. This vulnerability is dangerous for the cases when TLS compression is enabled. CRIME attack works by changing the properties of compression function and this way by discovering the size changes of the compressed message, and this way the original cooking can be learned. I will not describe the algorithm fully, because the compression algorithm is rather complicated to be fully described here. (Ritter 2012.)

TIME and BREACH vulnerabilities basically do the same, but use HTTP level compression which can reveal more sensitive data to the attackers than in the CRIME vulnerability described previously. In addition to that, these vulnerabilities are more difficult to avoid. (Holz & Sheffer & Saint-Andre 2015.)

5 TECHNOLOGIES USED IN THE MEASUREMENTS

The network architecture that is listed below will be used for all the measurements and same devices will be used as well. I decided to use Windows OS, as it is most widely-spread OS, and making tests with it is more practical. Another reason is that Windows has less open-source projects and finding a correct VPN solution is more complicated. I decided that it will be more helpful.

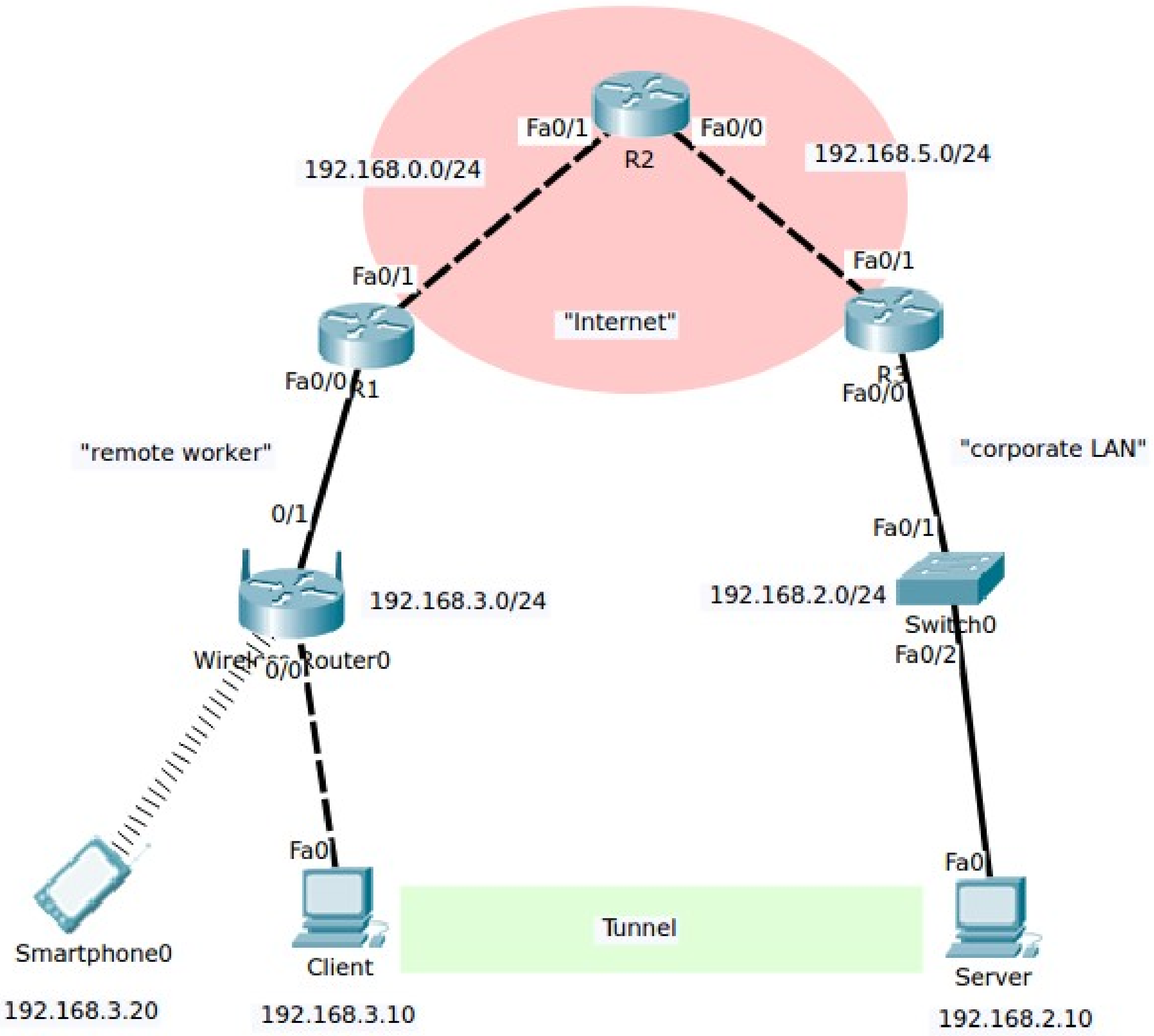


FIGURE 3 Network architecture

All the devices used in the test are listed below:

- End devices:

Test machines (2):

OS: Windows 8.1 Enterprise

Processor: Intel(R)Core(TM)i7-3770 CPU @ 3.40GHz

RAM: 16,0 GB

NIC: Realtek PCIe GBE Family Controller - Gigabit Controller

- Network devices:

Routers (3): Cisco 2811

Switches (2): Catalysg2960 Series

Used ports: FastEthernet

Cisco Adaptive Security Appliance 5505: Software Ver 8.2(5)

Wireless router: Cisco Small Business Wireless Access Point

The measurements will be done in different conditions, e.g. different VPN technologies will be used as well as no-encryption tests. The technologies are as follows: no encryption, client-based OpenVPN, OpenVPN ALS (Adito),IOS VPN. The main ideas of them will be described in the chapters below.

All the measurements will be done using the same tools in order to provide a fair comparison of the results. For the bandwidth test, I will use a client-server application called Iperf. The VPN client will serve as the Iperf client and the VPN server will be a Iperf server. The settings will be standard, and the test will be done several times in order to ensure that the results are repeating and are usual for those conditions.

After the measurements of a wired network I will carry out the tests in a wireless connection environment in order to see if the VPN technologies' performance will behave differently when used in a smartphone. The specifications of the smartphone and all the measurements will be made in Chapter 7.

5.1 Benchmark measurement

First, in order to get some benchmark results I will simply measure the speed of connection between PC1 (“client” in the Figure 3) and PC2 (“server”) without any encryption. In addition to that, it can serve as a test to see if the network was installed correctly and would produce realistic results.

5.2 Client based OpenVPN

The next test objective is OpenVPN. OpenVPN is a client-based VPN which requires a client to be installed and set up in both ends of the tunnel. Nowadays OpenVPN is available on a variety of different platforms: Linux, Solaris, OpenBSD, FreeBSD, NetBSD, Mac OS X, and Windows (OpenVPN Community 2015). In addition to that, OpenVPN can also be used on mobile platforms. Thanks to its comfortability and ease of use it is one of most popular VPN solutions nowadays.

5.4 Adito VPN

OpenVPN ALS (ADITO VPN) is practically the only one of the free browser-based solutions available in the market. Its history started as an SSL-explorer, but later the project was bought by Barracuda Networks and it became commercial, and Adito was developed as an open-source alternative. Adito is written in Java, compatible with Windows, Linux and it is easy to install. (McRee 2009). Adito is basically a browser-based OpenVPN. The main limitation it has is that fact that Adito does not forward UDP traffic. It can work only with TCP.

5.3 IOS VPN

Many VPN technologies can be implemented in Cisco ASA devices (5505 in our case), but this work will concentrate on SSL VPN solutions. Cisco offers different types of SSL VPN setup: Clientless SSL VPN (WebVPN), Thin-Client SSL VPN (Port Forwarding) and SSL VPN Client (SVC-Tunnel Mode). Clientless SSL VPN requires the client to have only a browser and that way access network resources. Thin-Client SSL VPN supports only the TCP protocol and requires downloading a Java applet from the browser and accessing the network this way. SSL VPN Client mode works similarly to OpenVPN – both the client and server need to have an application installed in order to be able to use the tunnel.

In my work I will use Cisco Thin-client technology which is also referred to as port forwarding. It requires an additional Java Applet to be installed to the client's machine. Basically this technology is an extension to WebVPN and allows a wider access to the network for applications that use static ports. Therefore, in order to install the connection between applications the administrator should set up port forwarding between ports that are used for the application. Typically this technology is used for access to POP3, SMTP, IMAP, SSH, and Telnet. (Cisco Systems Ltd 2008).

6 MEASUREMENTS

6. 1 Benchmark measurements

The first measurements are done without using any VPN tunnels, which makes it possible to see the real performance of the network in normal conditions. I will check the bandwidth between PC1 and PC2.

The first test was done with the TCP protocol, and the illustrations of the graph for 10 seconds' test and the results are below:

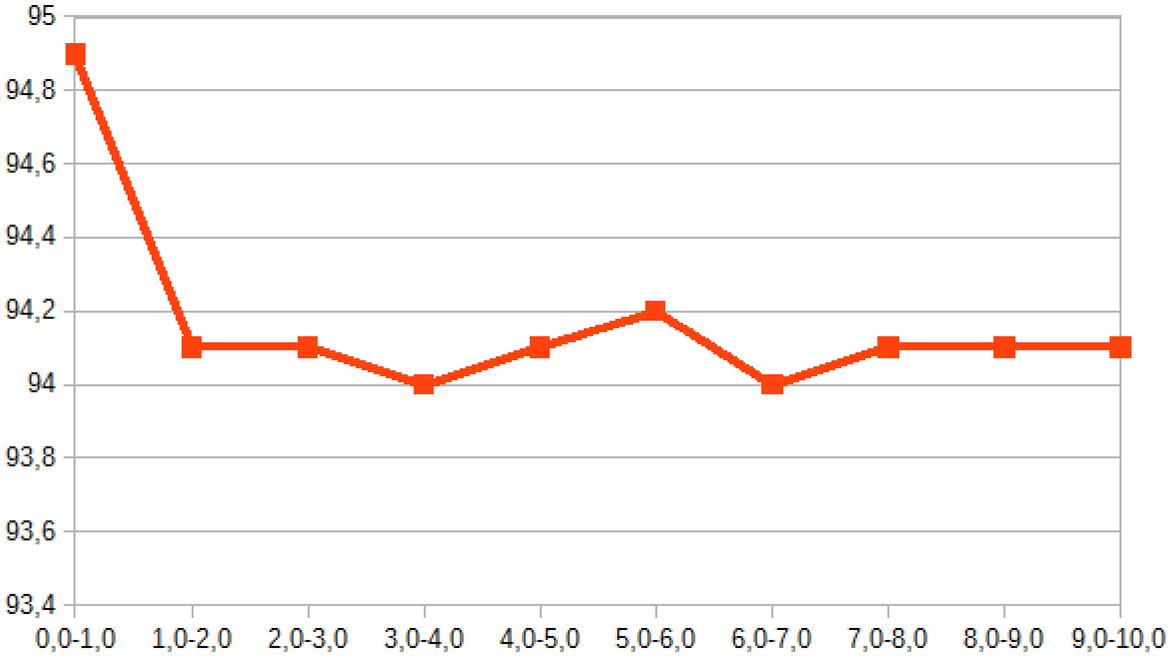


FIGURE 3 TCP graph, no encryption

The exact results are available in Table 4 below:

Interval (sec)	Transfer (Mbytes)	Bandwidth (Mbps)
0.0 – 1.0	11.3	94.9
1.0 – 2.0	11.2	94.1
2.0 – 3.0	11.2	94.1
3.0 – 4.0	11.2	94.0
4.0 – 5.0	11.2	94.1
5.0 – 6.0	11.2	94.2
6.0 – 7.0	11.2	94.0

7.0 – 8.0	11.2	94.1
8.0 – 9.0	11.2	94.1
9.0 – 10.0	11.2	94.1
0.0 – 10.0	112.1	94.18

TABLE 4 TCP result, no encryption

The next measurements were done for the UDP protocol. The graph and is listed below:

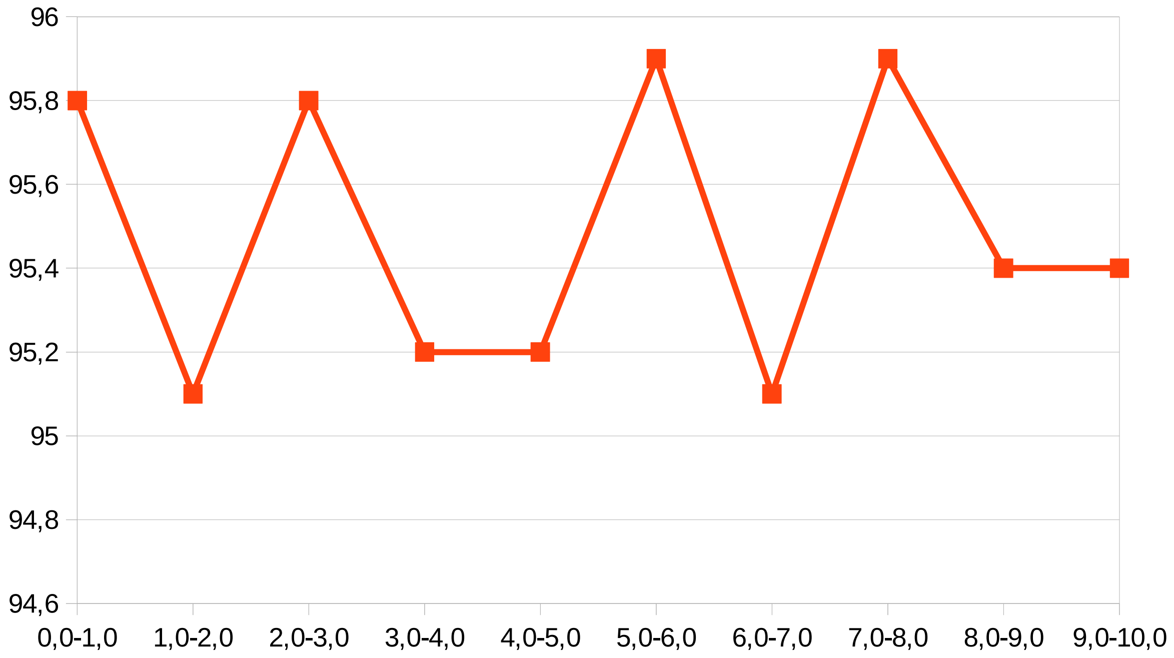


FIGURE 5 UDP graph, no encryption

The results are listed in Table 5:

Interval (sec)	Transfer (Mbytes)	Bandwidth (Mbps)
0.0 – 1.0	11.4	95.8
1.0 – 2.0	11.3	95.1
2.0 – 3.0	11.4	95.8
3.0 – 4.0	11.3	95.2
4.0 – 5.0	11.4	95.2
5.0 – 6.0	11.4	95.9
6.0 – 7.0	11.3	95.1
7.0 – 8.0	11.4	95.9

8.0 – 9.0	11.4	95.4
9.0 – 10.0	11.4	95.4
0.0 – 10.0	113.7	95.49

TABLE 5 UDP results, no encryption

According to the results of the measurements I calculated the average value and standard deviation. The results are presented in the table below:

	TCP	UDP
Average (Mbps)	94.18	95.49
Standard deviation (Mbps)	0.26	0.34

TABLE 6 No encryption results

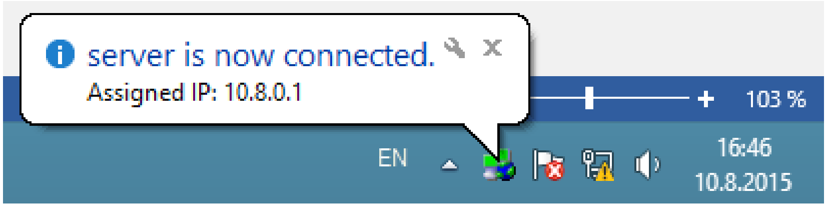
As it is visible in the results, the average TCP speed is 94.10Mbps and UDP is 95.50Mbps. As expected, the TCP bandwidth is a bit lower than UDP. Considering that 100Mbps is the maximum speed for our network (due to use of FastEthernet ports which improves the performance of Ethernet from 10Mbps to 100Mbps), the results are satisfactory and I can say that the network is installed correctly, because it produces trustworthy results.

6.2 OpenVPN installation and measurements

The next technology to be tested is OpenVPN. Its installation is relatively easy, the community provides a simple tutorial for installing the OpenVPN server and client (OpenVPN Community 2013). The tutorial includes downloading, creating certificates and keys, changing configuration and running it. I will not describe the details of the installation, because they are already precisely described in the community tutorial.

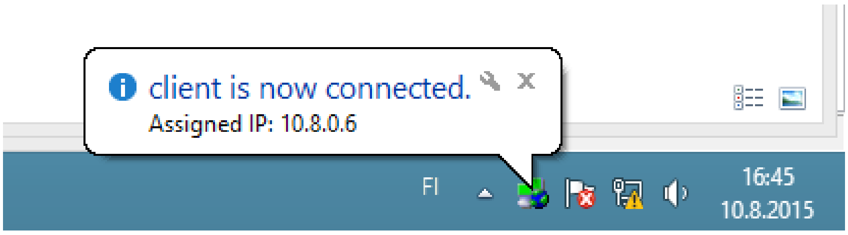
Installing OpenVPN on Windows has some of its problems and difficulties which are not met in other operating systems. Usually they are connected to access rights issues. For example, I had a problem while generating the keys. This problem was solved by generating the keys on a Linux machine. Another typical problem that OpenSSL has with Windows OS is the firewall. This problem is easily solved by adding exceptions to the firewall rules. Even though new Windows versions have many issues, OpenVPN has an active community so most of solutions are already described.

After the installation both the server and client are connected successfully and assigned IP addresses as shown in the pictures below:



server is now connected.
Assigned IP: 10.8.0.1

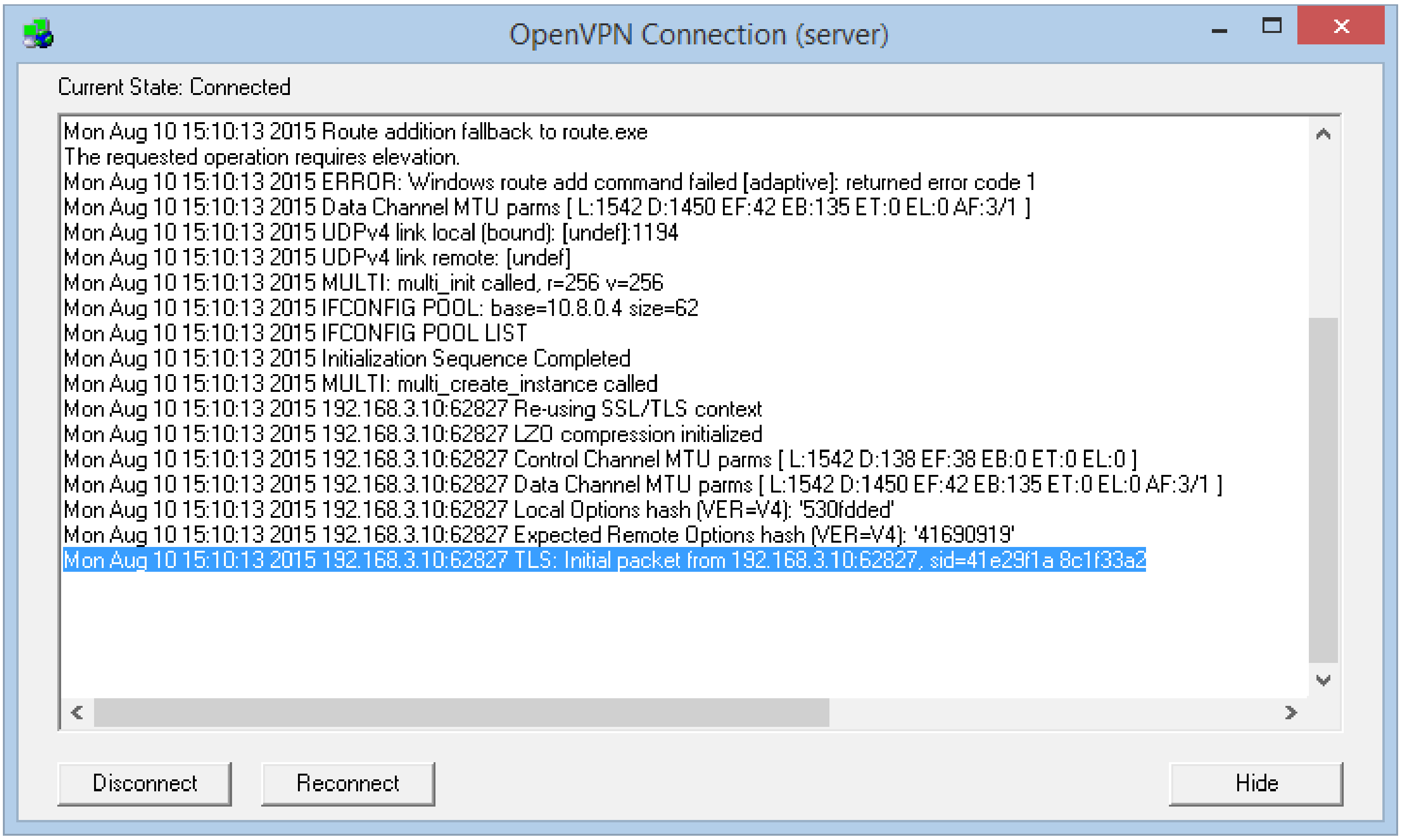
FIGURE 6 Client connected



client is now connected.
Assigned IP: 10.8.0.6

FIGURE 7 Server connected

After the server and client are connected it is possible to see the connection problems and logs through the applications on both sides. For example, the screenshot below shows the logs of the server which just received connection from the client:



OpenVPN Connection (server)

Current State: Connected

```
Mon Aug 10 15:10:13 2015 Route addition fallback to route.exe
The requested operation requires elevation.
Mon Aug 10 15:10:13 2015 ERROR: Windows route add command failed [adaptive]: returned error code 1
Mon Aug 10 15:10:13 2015 Data Channel MTU parms [ L:1542 D:1450 EF:42 EB:135 ET:0 EL:0 AF:3/1 ]
Mon Aug 10 15:10:13 2015 UDPv4 link local (bound): [undef]:1194
Mon Aug 10 15:10:13 2015 UDPv4 link remote: [undef]
Mon Aug 10 15:10:13 2015 MULTI: multi_init called, r=256 v=256
Mon Aug 10 15:10:13 2015 IFCONFIG POOL: base=10.8.0.4 size=62
Mon Aug 10 15:10:13 2015 IFCONFIG POOL LIST
Mon Aug 10 15:10:13 2015 Initialization Sequence Completed
Mon Aug 10 15:10:13 2015 MULTI: multi_create_instance called
Mon Aug 10 15:10:13 2015 192.168.3.10:62827 Re-using SSL/TLS context
Mon Aug 10 15:10:13 2015 192.168.3.10:62827 LZO compression initialized
Mon Aug 10 15:10:13 2015 192.168.3.10:62827 Control Channel MTU parms [ L:1542 D:138 EF:38 EB:0 ET:0 EL:0 ]
Mon Aug 10 15:10:13 2015 192.168.3.10:62827 Data Channel MTU parms [ L:1542 D:1450 EF:42 EB:135 ET:0 EL:0 AF:3/1 ]
Mon Aug 10 15:10:13 2015 192.168.3.10:62827 Local Options hash (VER=V4): '530fdded'
Mon Aug 10 15:10:13 2015 192.168.3.10:62827 Expected Remote Options hash (VER=V4): '41690919'
Mon Aug 10 15:10:13 2015 192.168.3.10:62827 TLS: Initial packet from 192.168.3.10:62827, sid=41e29f1a 8c1f33a2
```

FIGURE 8. Server logs

Even though the tunnel is set, the traffic will not come through it unless it is forwarded. However, by default all the traffic that is sent to the OpenVPN server is tunneled and encrypted. As PC2 (“server” in the architecture) is the server, I did not need to make any additional configurations, and the traffic was encrypted which can be seen in software like Wireshark. Then I can start testing the speed between the server and client,. The graph is listed below:

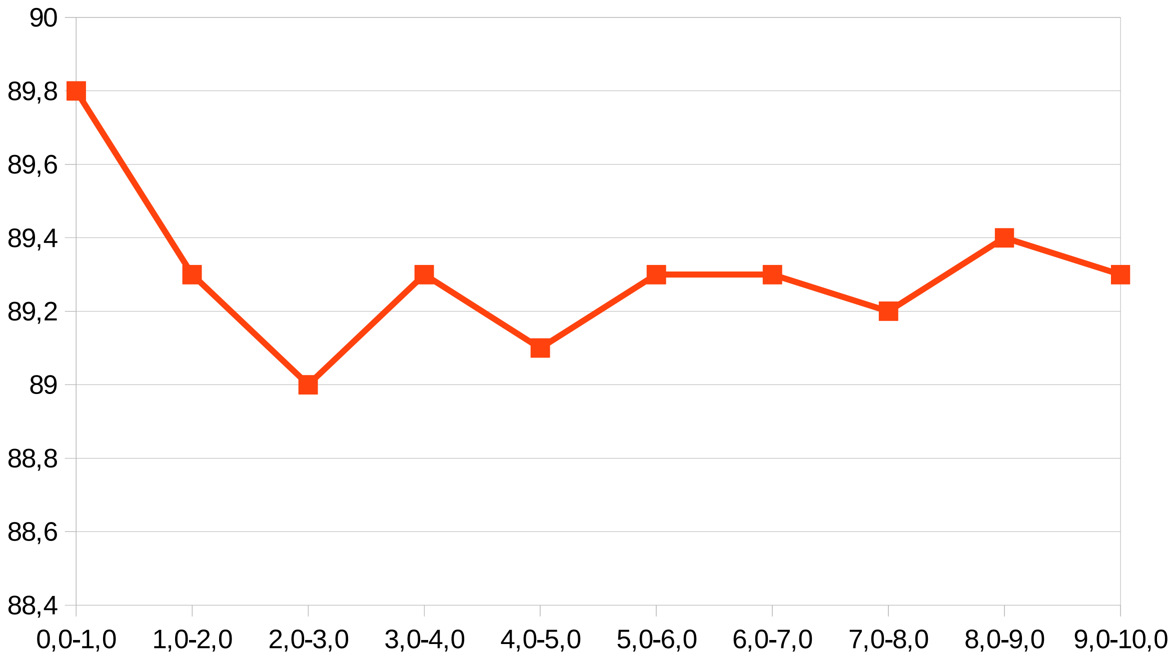


FIGURE 9 TCP graph, OpenVPN

The measurement results are listed in Table 7:

Interval (sec)	Transfer (Mbytes)	Bandwidth (Mbps)
0.0-1.0	10.7	89.8
1.0-2.0	10.6	89.3
2.0-3.0	10.6	89
3.0-4.0	10.6	89.3
4.0-5.0	10.6	89.1
5.0-6.0	10.6	89.3
6.0-7.0	10.6	89.3
7.0-8.0	10.6	89.2

8.0-9.0	10.7	89.4
9.0-10.0	10.6	89.3
0.0-10.0	106.2	89.3

TABLE 7 TCP result OpenVPN

The measurements for the UDP protocol are made in same way and shown in Figure 10:

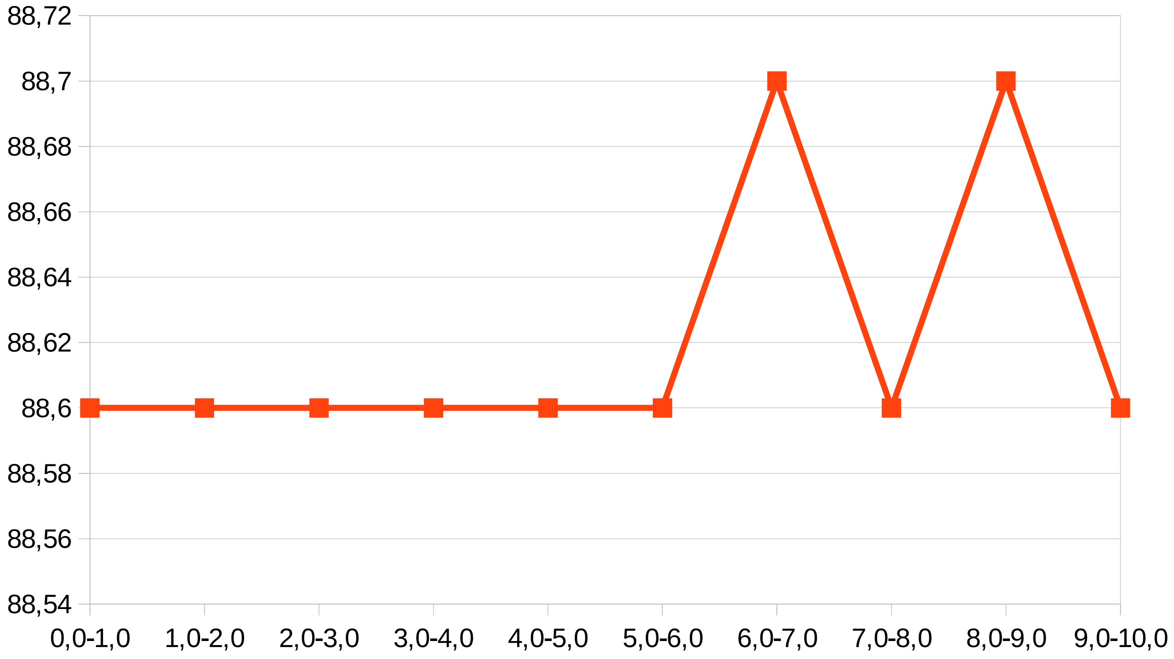


FIGURE 10 UDP graph, OpenVPN

The results of measurements of the UDP protocol are shown in Table 8:

Interval (sec)	Transfer (Mbytes)	Bandwidth (Mbps)
0.0-1.0	10.6	88.6
1.0-2.0	10.6	88.6
2.0-3.0	10.6	88.6
3.0-4.0	10.6	88.6
4.0-5.0	10.6	88.6
5.0-6.0	10.6	88.6
6.0-7.0	10.6	88.7
7.0-8.0	10.6	88.6
8.0-9.0	10.6	88.7

9.0-10.0	10.6	88.6
0.0-10.0	106	88.62

TABLE 8 UDP result, OpenVPN

From the received values I can calculate Standard deviation and Average bandwidth:

	TCP	UDP
Average (Mbps)	89.3	88.62
Standard deviation (Mbps)	0.21	0.04

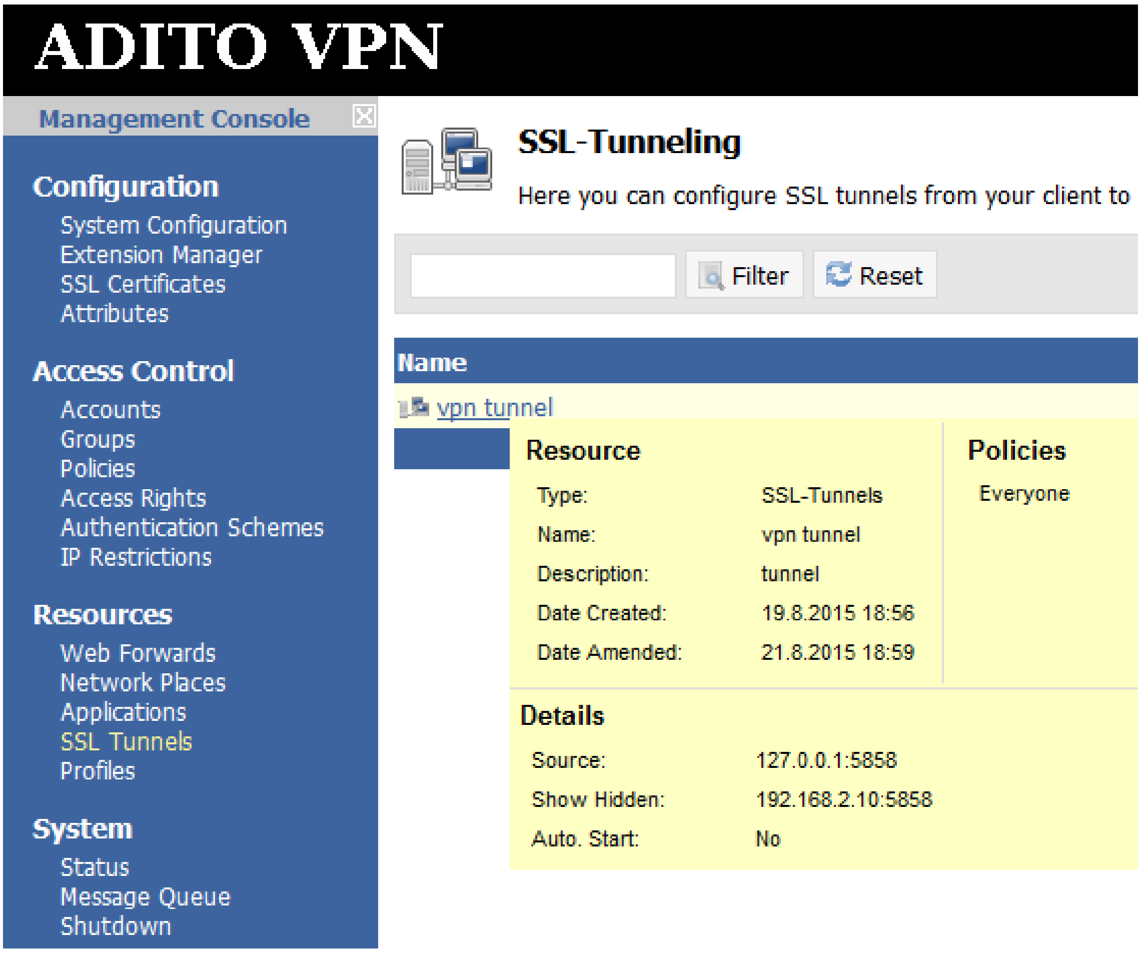
TABLE 9 OpenVPN results

Standard deviation is relatively small, so I can make a conclusion that the bandwidth is spread evenly, and there are not many unusual speed changes. The erformance of both protocols dropped in comparison to the benchmark measurements.

6.3 Adito installation and measurements

As mentioned in Chapter 5, I will use Adito VPN as the browser-based SSL VPN solution. Adito is written in Java so one of the requirements is having Java installed. Unfortunately, there are several issues with the compatible JRE and it is preferable to use 1.7. However, it is possible to use it on the latest JRE (1.8), and there are guides available. Earlier versions (<1.6) are not supported. In my installation I used the 1.7 version.

Adito installation guide that was written by the creator of the Adito project itself and explains most of the details which makes the installation easy (Werner 2011). However, installing the Adito server is not enough to be able to use VPN tunneling. After the installation of the Adito webserver is complete, it is possible to install different so-called applications. For example, there is an application used for remote desktop service. In this case I need to use the VPN tunnel, so I should create new one. The settings of the created tunnel are visible on the Figure 20 below:



SSL-Tunneling

Here you can configure SSL tunnels from your client to

vpn tunnel

Resource

Type: SSL-Tunnels
Name: vpn tunnel
Description: tunnel
Date Created: 19.8.2015 18:56
Date Amended: 21.8.2015 18:59

Policies

Everyone

Details

Source: 127.0.0.1:5858
Show Hidden: 192.168.2.10:5858
Auto. Start: No

FIGURE 11 Tunnel settings

After the Tunnel is created, it is advised to create new Adito user account for safety reasons because using only Admin account of remote computers might be dangerous. After that it is possible to access Adito web-server from any location, log in from user account and run the tunnel application. Adito agent will be downloaded and run.

Now, according to the tunnel settings, all the traffic will be redirected from 127.0.0.1:5858, e.g. all traffic leaving local PC from port 5858, to port 5858 of 192.168.2.10. In order to measure the performance, I should send the traffic to 127.0.0.1:5858 and later it will be redirected. And the results of the test are below:

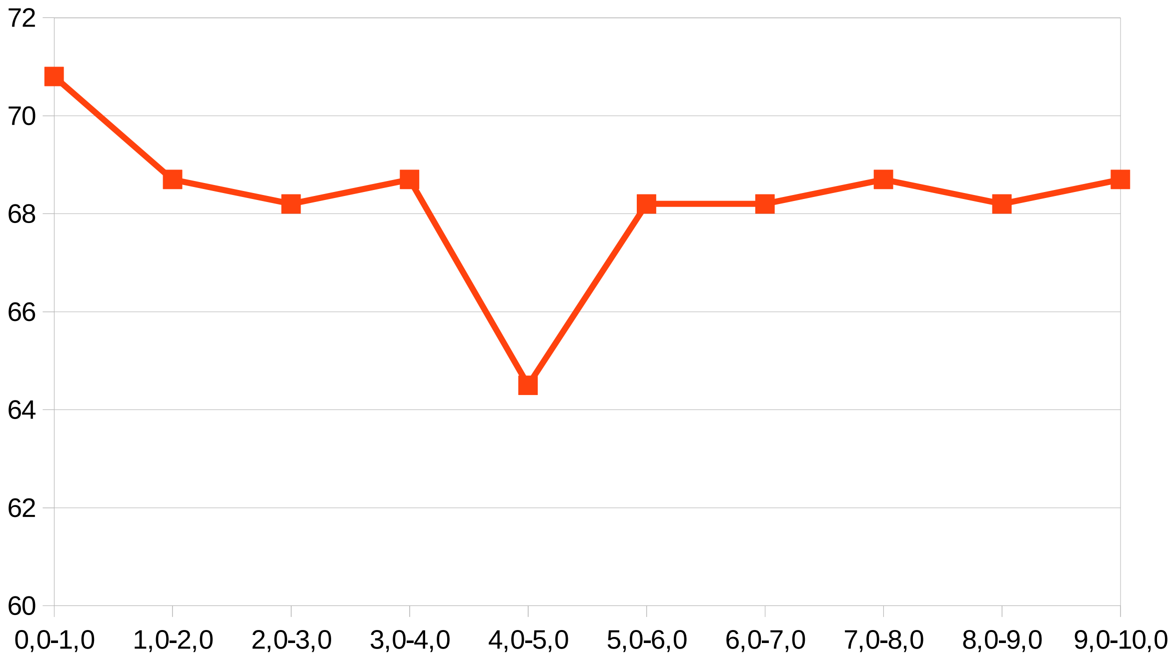


FIGURE 12 TCP graph, Adito

The numbers are shown in Table 10:

Interval (sec)	Transfer (Mbytes)	Bandwidth (Mbps)
0.0-1.0	8.44	70.8
1.0-2.0	8.19	68.7
2.0-3.0	8.13	68.2
3.0-4.0	8.19	68.7
4.0-5.0	7.69	64.5
5.0-6.0	8.13	68.2
6.0-7.0	8.13	68.2
7.0-8.0	8.19	68.7
8.0-9.0	8.13	68.2
9.0-10.0	8.19	68.7
0.0-10.0	81.41	68.24

TABLE 10 TCP result, Adito

Average result and standard deviation are available in Table 11 below:

	TCP	UDP

Average (Mbps)	68.24	-
Standard deviation (Mbps)	1.54	-

TABLE 11 Adito results

Adito supports only TCP forwarding so testing UDP protocol is impossible. By the results of TCP measurements it is possible to see that the speed dropped even more than in case with OpenVPN.

6.4 Cisco Thin-client SSL VPN

Cisco SSL VPN requires addition software – Cisco ASA device, so the architecture is a bit different and is presented below:

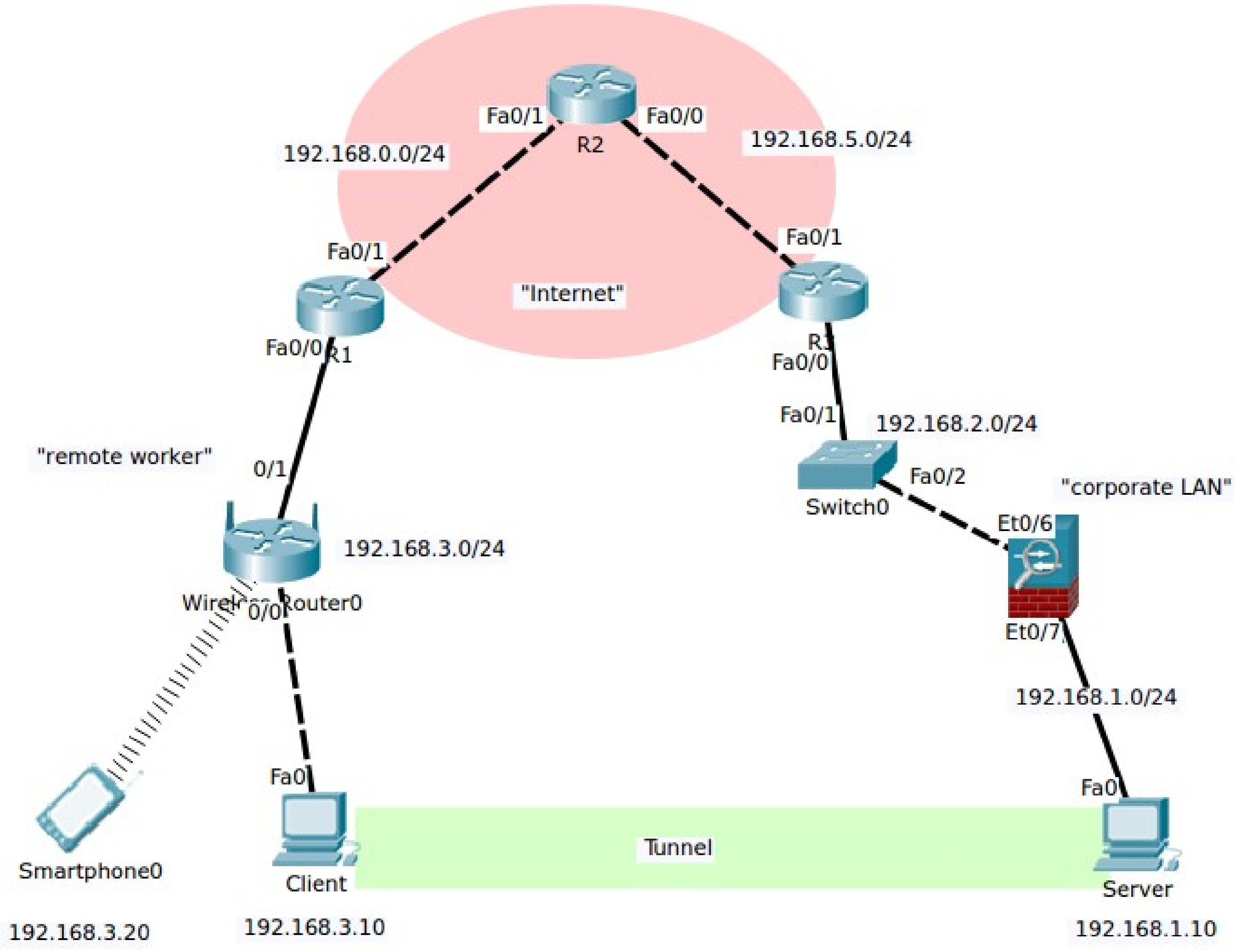


FIGURE 13 Cisco architecture

The installation process consists of the following steps:

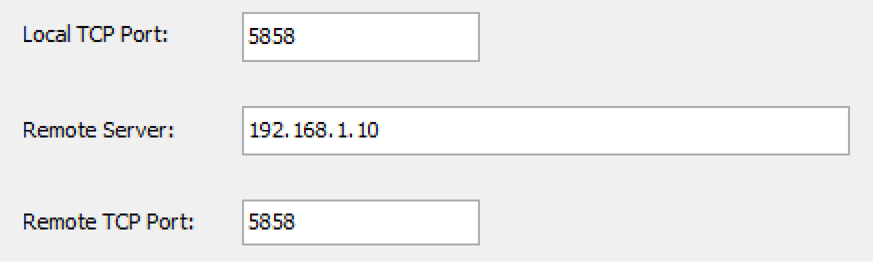
1) Configure WebVPN

First of all I configure WebVPN at the Cisco ASA device. The settings can be done either using ASDM interface (downloadable Java applet) or in console. There is plenty of easy to follow documentation on this issue (Cisco Systems Inc 2008) so I will not describe the steps here.

2) Configure port forwarding

After WebVPN is installed (it can be tested by trying to access and log in to the VPN server in the browser) port forwarding should be configured too.

The port forwarding settings are displayed in the Figure 25 below:



Local TCP Port: 5858

Remote Server: 192.168.1.10

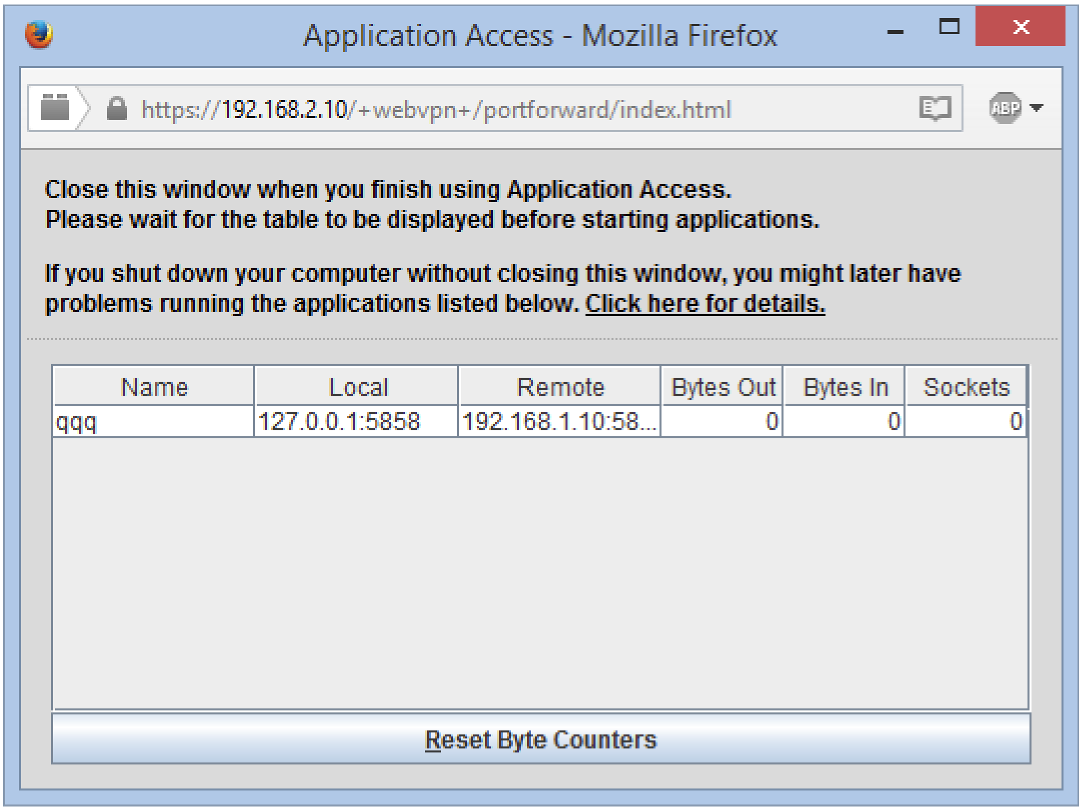
Remote TCP Port: 5858

FIGURE 14 Port forwarding configuration

3) Download Java applet on client side

Next time I log in to VPN server using the browser I can download the Java-based applet that will install to the PC and do the port forwarding. No additional configuration is required after this point.

The screenshot below shows the output of the downloaded applet and this output shows that the tunnel between ports is successfully established. From the screenshot it is also possible to see the statistics of usage:



Application Access - Mozilla Firefox
https://192.168.2.10/+webvpn+/portforward/index.html

Close this window when you finish using Application Access.
Please wait for the table to be displayed before starting applications.

If you shut down your computer without closing this window, you might later have problems running the applications listed below. Click here for details.

Name	Local	Remote	Bytes Out	Bytes In	Sockets
qqq	127.0.0.1:5858	192.168.1.10:58...	0	0	0

FIGURE 15 Applet interface

Testing can be done by using the local PC's configured forwarding port. In my case I tried accessing that port by Iperf. The server's Iperf application was successfully receiving the traffic which means that port forwarding was installed successfully. So the client's command to the Iperf looked this way:

```
bin/iperf.exe -c 127.0.0.1 -P 1 -i 1 -p 5858 -f m -t 10
```

and the results of this command and the table of the measurement results are below:

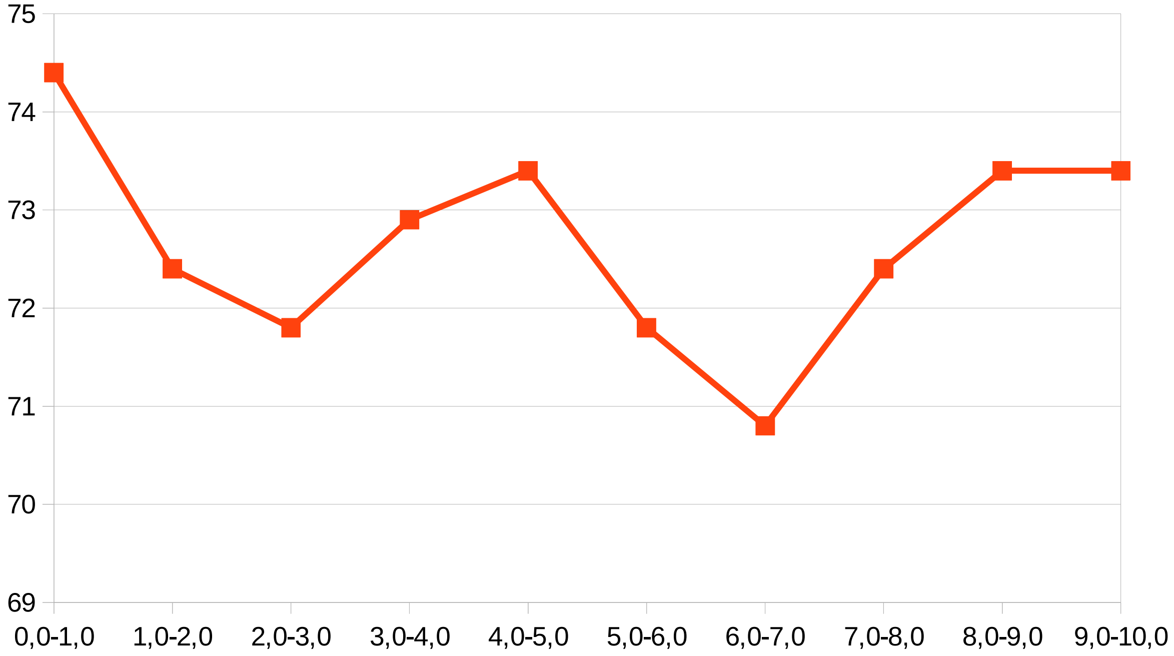


FIGURE 16 TCP graph, Thin-Client SSL VPN

The numbers of the TCP measurement are available in the table below:

Interval (sec)	Transfer (Mbytes)	Bandwidth (Mbps)
0.0-1.0	8.88	74.4
1.0-2.0	8.63	72.4
2.0-3.0	8.56	71.8
3.0-4.0	8.69	72.9
4.0-5.0	8.75	73.4
5.0-6.0	8.56	71.8
6.0-7.0	8.44	70.8
7.0-8.0	8.63	72.4
8.0-9.0	8.75	73.4
9.0-10.0	8.75	73.4
0.0-10.0	86.64	72.59

TABLE 12 TCP result, Thin-Client SSL VPN

And the average value and standard deviation are shown in Table 13:

	TCP	UDP

Average (Mbps)	72.59	-
Standard deviation (Mbps)	1.04	-

TABLE 13 Thin-Client SSL VPN results

The results of these measurements will be discussed in more detail on Chapter 8.

7 MOBILE MEASUREMENTS

In addition to measurements with PCs I will also measure the performance for smartphone connected wirelessly. This will help understand how the VPNs perform on the device and how easy they are to be used remotely. Nowadays as the role of smartphones is growing every day, the performance of VPNs on mobile devices has become as important for many remote workers as the VPNs performance on PCs. Smartphones have lower performance then PCs so comparing the wireless results can give us more clearly defined results than measurements from the previous chapter.

7.1 Technical information

For the measurements of mobile performance Android device will be used. The smartphone specifications:

Model	LG Spirit 4G LTE H440N
OS	Android OS, v5.0.1 (Lollipop)
CPU	Quad-core 1.2 GHz Cortex-A53 - H440N
Chipset	Qualcomm MSM8916 Snapdragon 410
MEMORY	8 GB, 1 GB RAM

TABLE 14 Phone specifications

According to the device specifications, it is usual middle-class device with more or less average performance, so I can expect most of the devices to behave this way.

7.2 Benchmark measurements for mobile devices

The results were measured between the mobile device and the remote server. As in previous examples the data was collected using Iperf which is supported by Android and the results are available in the figures and tables below:

Interval (sec)	Transfer (Mbytes)	Bandwidth (Mbps)
0.0 – 1.0	2.64	22.2
1.0 – 2.0	3.05	25.6
2.0 – 3.0	3.38	28.3

3.0 – 4.0	3.44	28.8
4.0 – 5.0	3.06	25.7
5.0 – 6.0	3.41	28.6
6.0 – 7.0	3.15	26.4
7.0 – 8.0	2.98	25
8.0 – 9.0	2.89	24.3
9.0 – 10.0	3	25.1
0.0 – 10.0	31	26.01

TABLE 15 TCP performance table

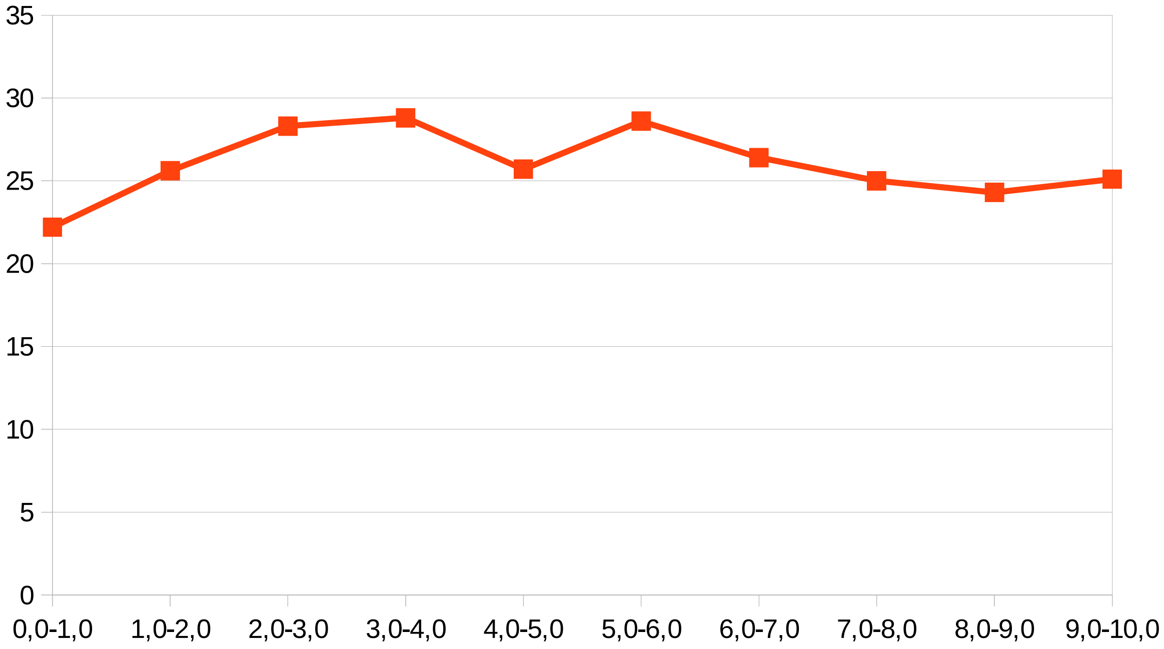


FIGURE 17 TCP performance graph

Accordingly the results for the UDP protocol are shown in Table 16 and Figure 18:

Interval (sec)	Transfer (Mbytes)	Bandwidth (Mbps)
0.0 – 1.0	4.04	33.9
1.0 – 2.0	4	33.5
2.0 – 3.0	3.94	33.1
3.0 – 4.0	4.01	33.6

4.0 – 5.0	3.98	33.4
5.0 – 6.0	4	33.5
6.0 – 7.0	4.04	33.9
7.0 – 8.0	3.99	33.5
8.0 – 9.0	4.02	33.8
9.0 – 10.0	4.03	33.8

TABLE 16 UDP performance table

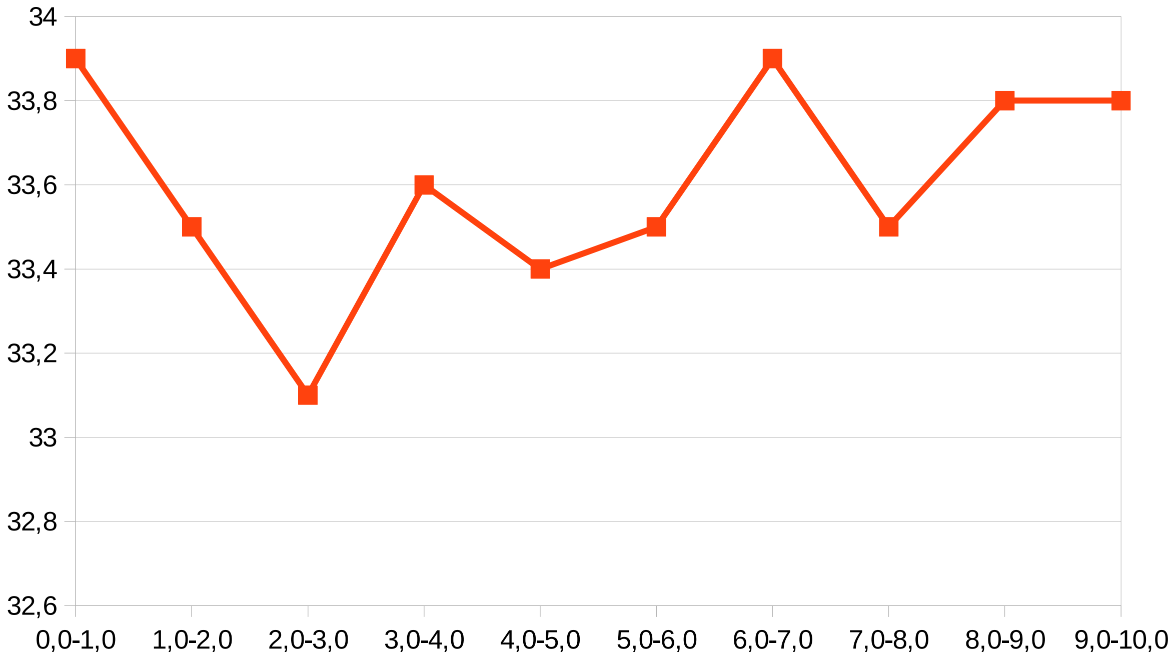


FIGURE 18 UDP performance graph

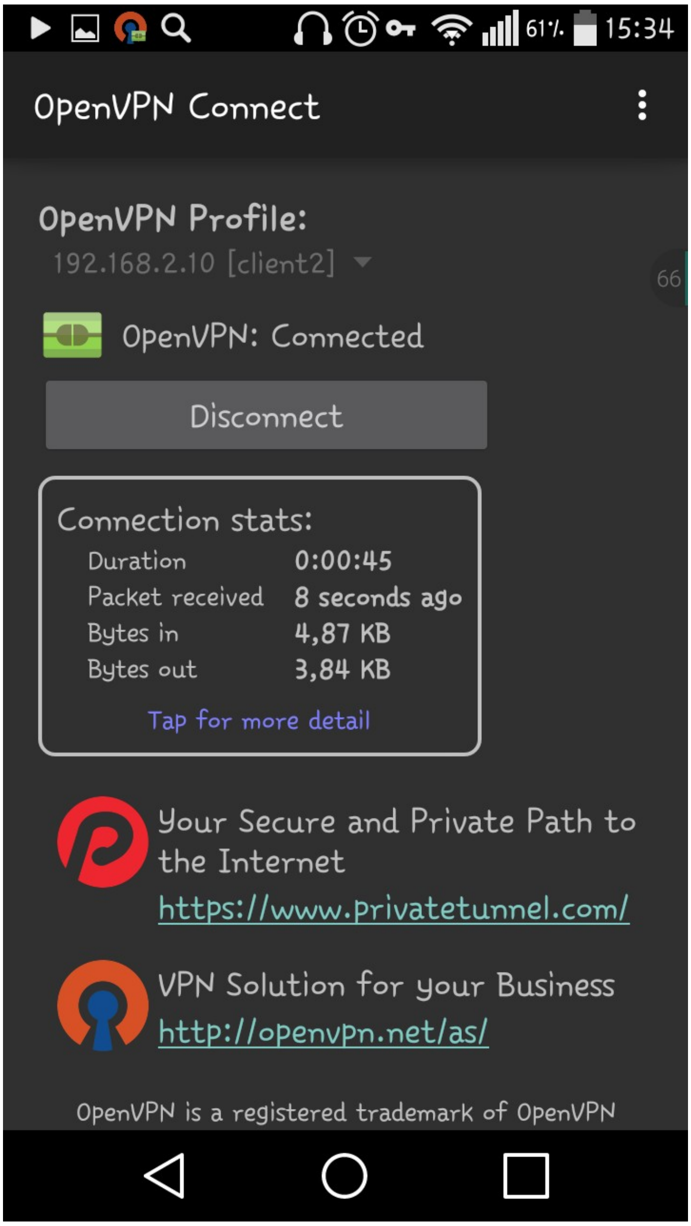
Out of all these results I can calculate standard deviation and average bandwidth:

	TCP	UDP
Average (Mbps)	26.1	33.58
Standard deviation (Mbps)	2.09	0.25

TABLE 17 Mobile benchmark results

7.2 OpenVPN mobile performance

OpenVPN has easy to install and use official Android application such as shown in the screenshot below:



15:34
61%
OpenVPN Connect
66
Disconnect
Tap for more detail
Your Secure and Private Path to the Internet
https://www.privatetunnel.com/
VPN Solution for your Business
http://openvpn.net/as/
OpenVPN is a registered trademark of OpenVPN

OpenVPN Profile:
192.168.2.10 [client2]

OpenVPN: Connected

Connection stats:

Duration	0:00:45
Packet received	8 seconds ago
Bytes in	4,87 KB
Bytes out	3,84 KB

FIGURE 19 OpenVPN Android application

The installation and the use of it are very easy – the client configuration files first should be moved to the device and then the application should be pointed to them. After that I used Android version of Iperf to make the measurements:

Interval (sec)	Transfer (Mbytes)	Bandwidth (Mbps)
0.0-1.0	2.64	22.2
1.0-2.0	3.05	25.6
2.0-3.0	3.38	28.3

3.0-4.0	3	25.1
4.0-5.0	3.06	25.7
5.0-6.0	3.41	28.6
6.0-7.0	3.15	26.4
7.0-8.0	2.98	25
8.0-9.0	2.89	24.3
9.0-10.0	3	25.1

TABLE 18 TCP performance table, mobile OpenVPN

The graph for the numbers above from Table 18 is shown below:

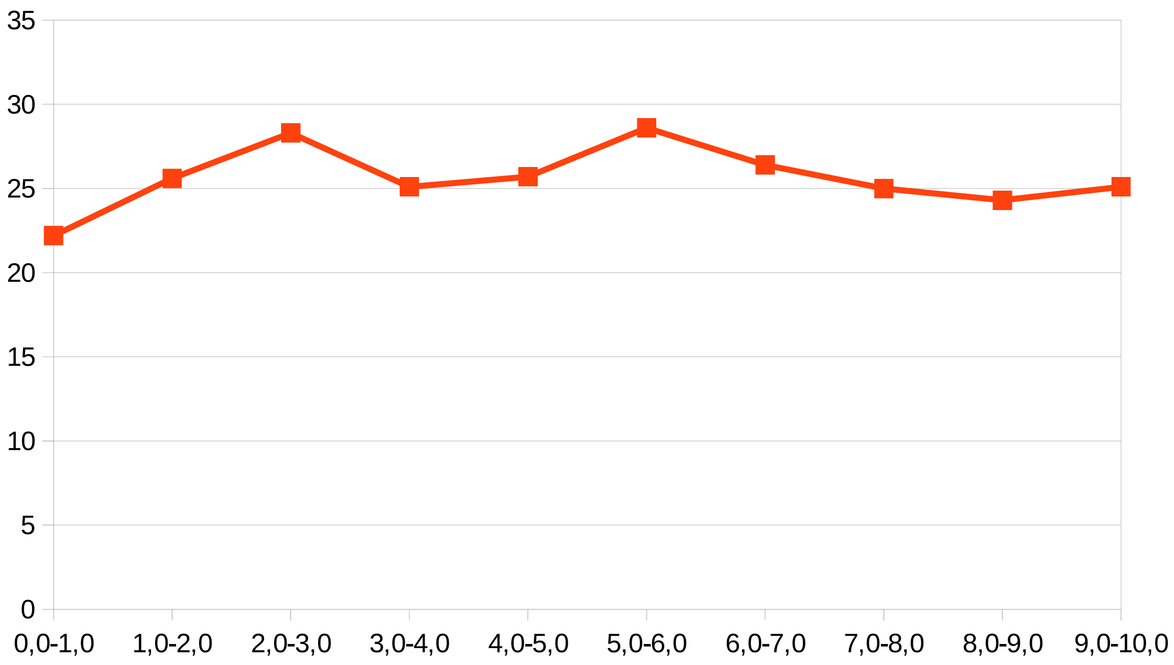


FIGURE 20 TCP performance graph, mobile OpenVPN

Then I put the USP results in the table in similar way:

Interval (sec)	Transfer (Mbytes)	Bandwidth (Mbps)
0.0-1.0	3.66	30.7
1.0-2.0	3.82	32.1
2.0-3.0	4	33.5
3.0-4.0	4.01	33.6
4.0-5.0	3.67	30.5

5.0-6.0	4	33.5
6.0-7.0	4.02	33.8
7.0-8.0	4.03	33.8
8.0-9.0	4.04	33.9
9.0-10.0	4.03	33.8

TABLE 19 UDP performance table, mobile OpenVPN

And make the graph for the data from Table 19:

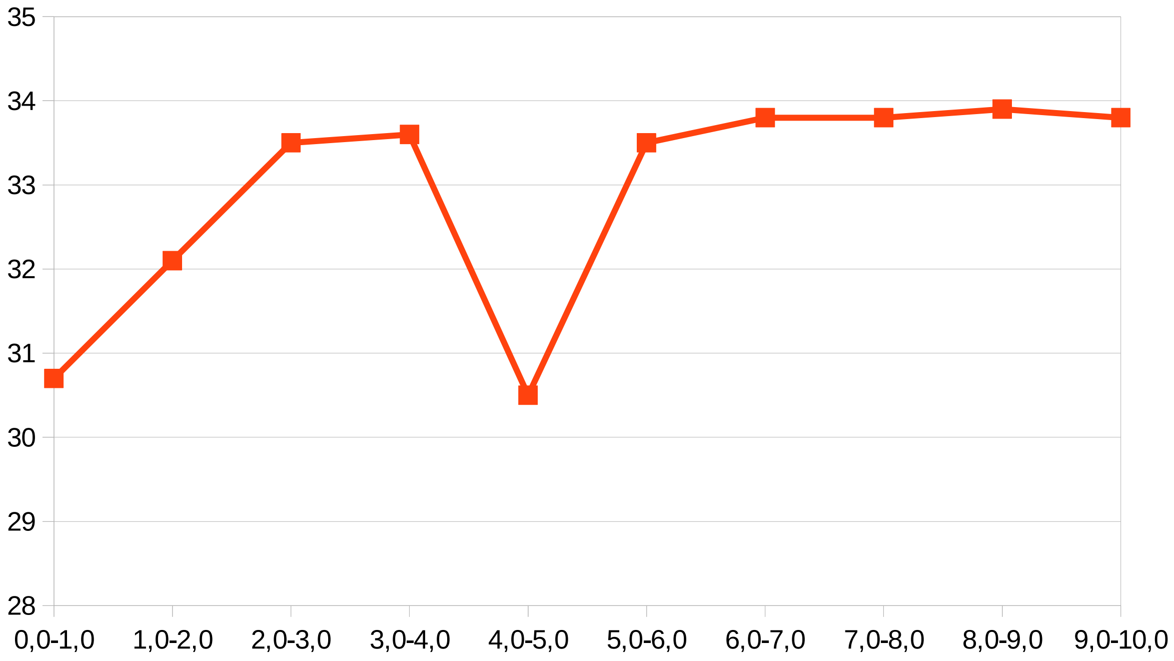


FIGURE 21 UDP performance graph, mobile OpenVPN

Out of all these results I can calculate standard deviation and average bandwidth:

	TCP	UDP
Average (Mbps)	25.59	32.82
Standard deviation (Mbps)	1.86	1.33

TABLE 20 Mobile OpenVPN results

7.3 Cisco Thin-client SSL mobile performance

As it is stated in the Cisco documentation (Cisco 2015) “We do not provide clientless VPN support for Java, auto applet download, smart tunnels, plug-ins, port forwarding, and e-mail proxy for mobile devices” , e.g. using and measuring Thin-client VPN for mobile platform (Android in our case) is impossible.

8 COMPARING THE MEASUREMENT RESULTS

8.1 Comparing the performance of wired connection

Measuring the bandwidth of different VPN technologies in the wired environment (PC to PC) gave us quite different result. The graph showing the differences is available below:

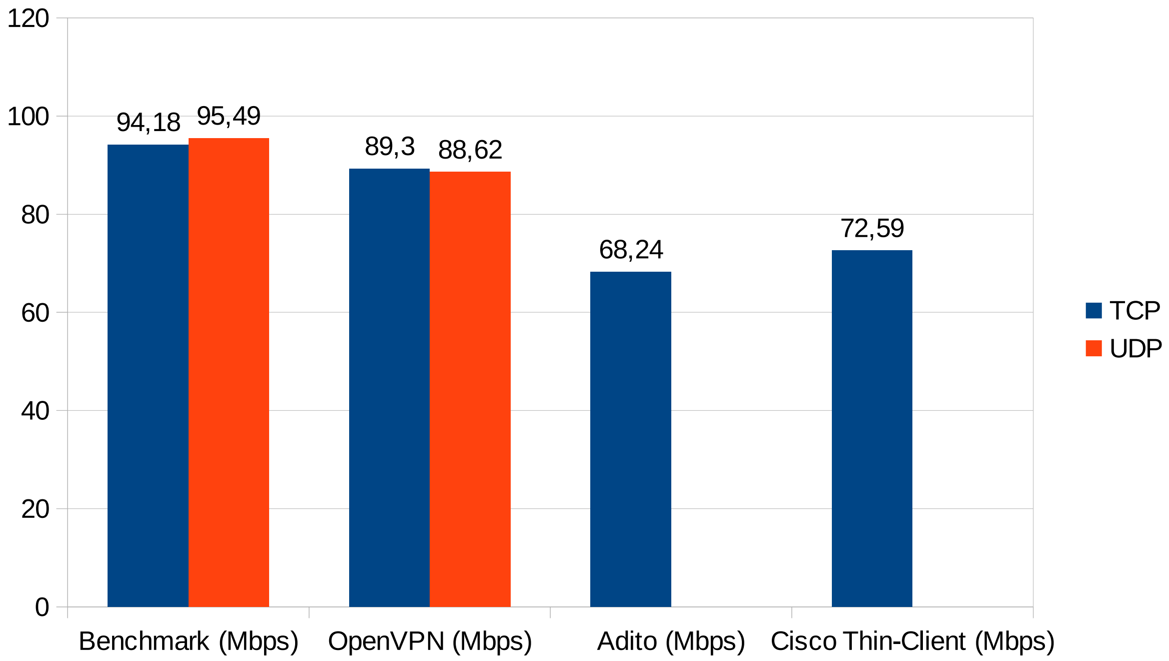


FIGURE 22 Difference in bandwidth of wired connection

As I can see, the best performing results are the benchmark results. The results were made in order to see how the real-world measurements would differ from the maximum bandwidth of the network (100 Mbps). The speed dropped to approximately 95 Mbps, which is normal result and means that there are no big problems in the architecture.

The next best-performing results belong to OpenVPN which shows very high speed and is just 5,2% (if I compare only TCP performance) lower than the benchmark test.

The next in the list of best-performing technologies is Cisco Thin-client with 72.59 Mbps which is 23% lower than the benchmark result and 18,8% lower than OpenVPN result.

And the last in the list is Adito VPN with 68,24 Mbps and its bandwidth drop is 27,6% comparing to benchmark and 6% comparing to the Cisco Thin-client.

8.2 Comparing the performance of wireless smartphone VPNs.

Unfortunately, as many of the technologies that I used do not support mobile devices, there is not enough material for the proper comparison. The graph below shows the difference in performance of our measurements:

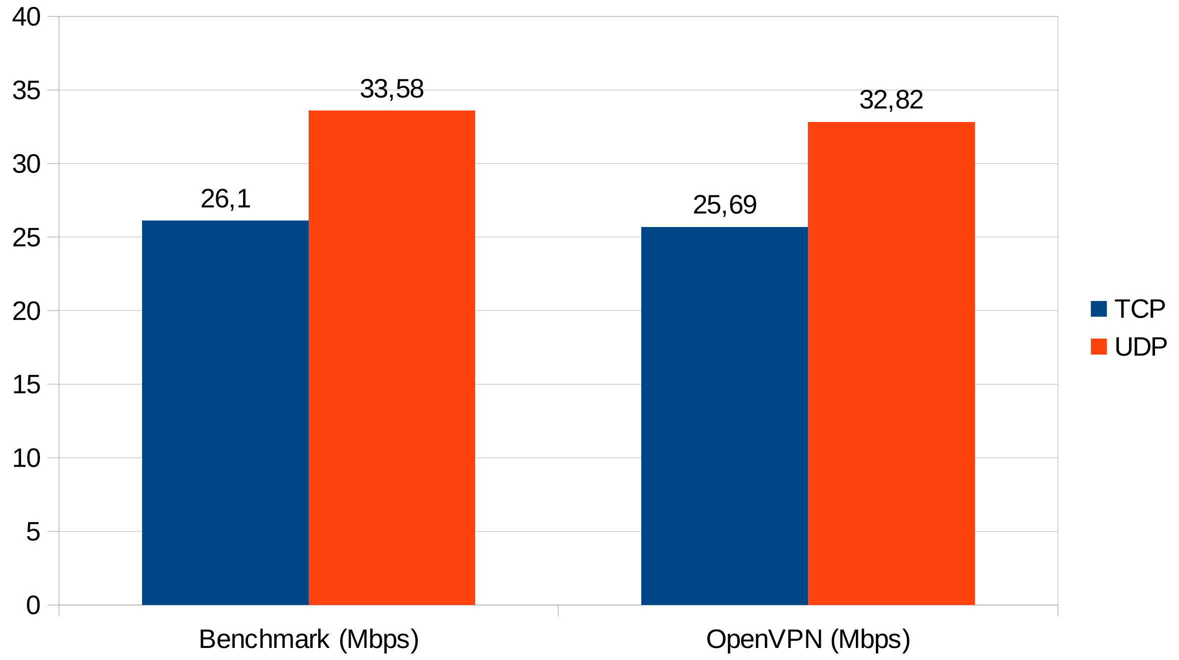


FIGURE 23 Difference in bandwidth of wireless connection

Interesting fact is that the speed drop of OpenVPN in mobile environment is very low. The drop of TSP and UDP is only 2,3% and 1,6% accordingly. In some other measurements the difference was sometimes even smaller, so I can make a conclusion that the performance of mobile OpenVPN is very good and does not mean big bandwidth drop. In order to make the measurements more trustworthy the mobile device was always placed at the same spot and I tried to avoid causing any interference.

9 CONCLUSION

In my work I studied the working principle of different VPN technologies, their main applications, strengths and weaknesses and possible security issues. Based on all the information I chose several most popular SSL/TLS VPN products, described their work, installation, and measured the performance. The idea of the thesis was in finding the optimal VPN solution which is easy to maintain, provides high security and gives the best performance. In order to find this solution I ran several performance tests and studied the possible issues of these technologies.

In short the best performing technology turned out to be OpenVPN. It has many advantages over other options that I tried – it shows very good performance, supports a mobile version, is supported by a wide community and is open-source. Also, as the measurements of the mobile version I discovered that the performance drop is very small, which is a good advantage as well. In addition to that, OpenVPN was the easiest technology to install and implement. I would suggest OpenVPN to be used by people who care about their privacy and small businesses. Bigger businesses can also find many advantages in OpenVPN unless they want some additional security features.

The next technology that shows good performance results is Cisco Thin-Client. Cisco security appliance that was used for the test is also usable for a huge variety of other applications, and it also supports other types of VPNs as well. It is impossible to make the conclusions about the usability of Cisco Security Appliance based only on tests of Thin-Client VPN. However, it is possible to make some conclusions about the usability of the selected VPN technology alone. Cisco Thin-client is easy to maintain, install, provides quite good performance and is a good option. However, I would advice it to be used in other environments than OpenVPN. For example, this is a very good solution for corporate networks of businesses. The reason for this is first of all in the price of the security appliance which is rather high, and can be affordable only for businesses. Secondly, the security appliance provides a wide range of additional security features which is going to be very useful for the corporate network.

And the last performance results are shown by Adito VPN. Among all of the solutions I would call it the least preferable one, even though it is quite often recommended as good solution. The installation is not very simple, the performance is low and the UDP protocol is not supported.

As the conclusion to the above thesis I would say that the most preferable VPN technology among the popular and up-to-day solutions nowadays is OpenVPN.

BIBLIOGRAPHY

Avinash, SM 2015. SSL Stripping for Newbies. WWW-publication. https://www.linkedin.com/pulse/ssl-stripping-newbies-avinash-sm Referred 5.11.2015.

Cisco Systems, Inc. 2015. Supported VPN Platforms, Cisco ASA 5500 Series. (pp. 5)

Cisco Systems, Inc. 2008. Thin−Client SSL VPN (WebVPN) IOS Configuration Example with SDM. (pp. 1). WWW-document. http://www.cisco.com/image/gif/paws/70664/IOSthinclient.pdf Referred 15.10.2015.

Ferguson, Paul & Huston, Geoff 1998. What is a VPN?

Gunter, Manuel 1998. Virtual Private Networks over the Internet. citeseer. com document, 1-7.

Frankel, Sheila et al. 2005. Guide to IPsec VPNs. NIST Special Publication : 800-77

Fluhrer, Scott, Mantin, Itsik, & Shamir, Adi 2001. Weaknesses in the key scheduling algorithm of RC4. In Selected areas in cryptography (pp. 1-24). Springer Berlin Heidelberg.

Georgiev, Martin, Iyengar, Subodh, Jana, Suman, Anubhai, Rishita, Boneh, Dan, & Shmatikov, Vitaly 2012. The most dangerous code in the world: validating SSL certificates in non-browser software. In Proceedings of the 2012 ACM conference on Computer and communications security (pp. 38-49). ACM. [Accessed 7 Nov. 2015].

Holz, Ralph, Sheffer, Yaron, & Saint-Andre, Peter 2015. Summarizing Known Attacks on Transport Layer Security (TLS) and Datagram TLS (DTLS). No. RFC 7457.

Lewis, Mark 2006. Comparing, Designing, and Deploying VPNs, Adobe Press: 12-19

Microsoft Corporation 2003. How SSL/TLS works. WWW-document. https://technet.microsoft.com/en-us/library/cc783349(v=ws.10).aspx Referred 14.9.2015.

Microsoft Corporation 2007. The MS-CHAP version 1 authentication protocol has been deprecated in Windows Vista. WWW-document. https://support.microsoft.com/en-au/kb/926170 Referred 4.10.2015.

Microsoft Corporation 2012. Microsoft Security Advisory 2743314. WWW-document. https://technet.microsoft.com/en-au/library/security/2743314 Referred 4.10.2015.

McRee, Russ 2009. Adito: Open-source, browser-based SSL VPN. ISSA Journal, 32–34.

McKinley, Holly Lynne 2003. SSL and TLS: A Beginners' Guide SANS Institute.

Mironov, Ilya 2002. (Not so) random shuffles of RC4. In Advances in Cryptology—CRYPTO 2002 (pp. 304-319). Springer Berlin Heidelberg.

Oracle Corporation 2010. Sun Directory Server Enterprise Edition 7.0. WWW-document. http://docs.oracle.com/cd/E19424-01/820-4811/aakhb/index.html Referred 1.10.2015.

OpenVPN Community 2015. Overview of OpenVPN. WWW-document. https://community.openvpn.net/openvpn/wiki/OverviewOfOpenvpn Referred 10.9.2015.

OpenVPN Community 2013. Easy Windows Guide. WWW-document. https://community.openvpn.net/openvpn/wiki/Easy_Windows_Guide#DownloadingandInstallingOpenVPN Referred 15.9.2015.

OpenSSL. Frequently asked questions. WWW-document. https://www.openssl.org/docs/faq.html Referred 17.9.2015

Patel, Baiju, Aboba, B., Dixon, W., Zorn, G., & Booth, S. 2001. Securing L2TP using IPsec. No. RFC 3193)

Ritter, Tom 2012. Details on the "Crime" Attack. WWW-document. https://www.nccgroup.trust/us/about-us/newsroom-and-events/blog/2012/september/details-on-the-crime-attack/ Referred 7.11.2015

Shea Richard 2000. L2TP: implementation and operation. Addison-Wesley Professional.

Scott, Charlie, Wolfe, Paul, & Erwin, Mike 1999 Virtual private networks. O'Reilly Media, Inc.

Thomas, Stephen 2000. SSL and TLS essentials. p.3 New Yourk.

Townsend, Kevin 1998. Understanding VPNs And PPTP. PC Network Advisor.

Turner, Sean, Polk, Tim 2011. Prohibiting secure sockets layer (SSL) version 2.0. Internet Engineering Task Force

VPN Technologies 2008. Definitions and Requirements, VPN Consortium

Werner, Lars 2011. Adito guide: Get RDP everywhere with Adito! WWW-publication. http://lars.werner.no/?p=640 Referred 26.9.2015