

Performance Evaluation of Software Virtual Private Networks (VPN)

C. Javier Castro Peña and Joseph Evans
Information & Telecommunication Technology Center
Department of Electrical Engineering & Computer Science
The University of Kansas, Lawrence, KS, 66045
[jcaspen,evans]@ittc.ukans.edu

Abstract

Virtual Private Networks implemented in software provide an economic and accessible alternative to hardware VPN solutions. Software VPNs may have a significant impact on performance, producing high CPU usage and limiting network throughput. This paper presents the performance measurements of several VPN programs. The results over a 100 Mb/s Ethernet link show that the transference speed can degrade in more than 65% while the CPU usage can reach 97%, when strong encryption is enabled. Compression implemented at the user level adds an additional CPU overhead that has a negative effect on the performance. The results over a low speed serial link show that the CPU usage is not significantly affected by the VPN. Furthermore, compression can be enabled without overhead, increasing the network throughput when the data is compressible.

1. Introduction

A VPN allows use of the Internet and other public networks to extend the reach of a local network. Although the performance of hardware devices that implement these mechanisms is well documented, the overhead added by software VPNs is not yet fully known.

This paper presents the performance observed on software VPN in terms of network throughput and CPU usage. Two main cases were studied: a fast network (100 Mb/s) and a slow network (10.3 kb/s).

2. Test environment and tools

In the fast network case, endpoints were connected through a 100 Mb/s switched network. The workstations were Pentium 233 Mhz, 128 Mb RAM PCs, using Linux kernel 2.2.10 in most cases and 2.0.36 with FreeS/Wan.

In the slow network case, endpoints were connected with 33.6 kb/s through the Internet. The dialup host was an AMD K6 400Mhz, 128 Mb PC running Linux 2.2.10.

Ttcp and the Unix command top were used for measurement. ttcp times the transmission and reception of data between two systems using the UDP and TCP protocols. It fills a memory buffer with data and then transmits.

3. VPN software tested

A number of different packages were evaluated. CIPE v1.3 implements 128 bit Blowfish encryption. FreeS/Wan v1.0 implements IPsec [1]. PoPToP v0.9.16 is a PPTP server and pptp-linux v1.0.2 a client. Tinc v0.3 and Tunnel Vision v1.0 use the Ethertap interface and use 128 bit encryption. Vpnd v1.0.8 implements 576 bit Blowfish over SLIP. Vtun v1.5b provides weak encryption (XOR).

4. VPN performance over a fast network

In this case, the point to point throughput was 10940 kB/s. Figure 1 shows the drop caused by the VPN.

When encryption and compression are enabled, performance decreases due to the higher CPU usage.

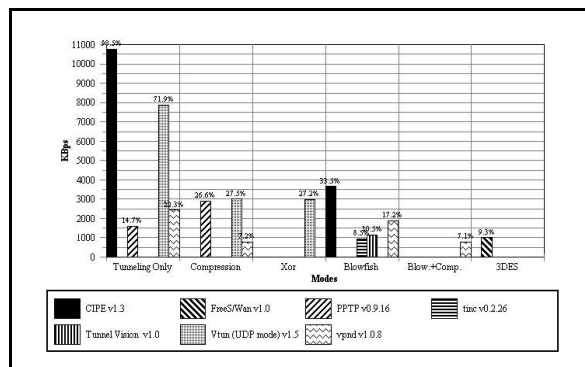


Figure 1 – VPN network throughput

Code optimizations can have an important effect in the performance. The encryption routines of Vpnd are implemented in assembly and fit in the L1 cache of a Pentium processor. There is almost no difference in the use of 1 bit and 576 bit keys, as table 1 shows.

Table 1 – Influence of Key Length in Vpnd

No encryption	1 bit encryption	576 bit key enc.
2431.52 kB/s (100%)	1852.50 kB/s (76.19%)	1851.30 kB/s (76.14%)

PPTP improves the network throughput when compression is enabled. The PPP driver, implemented at the kernel level, is in charge of providing the compression mechanism to PPTP. It seems to be more efficient than the user level implementations of the other programs.

The drop in network throughput is related to the increase of CPU usage shown in figures 2 and 3.

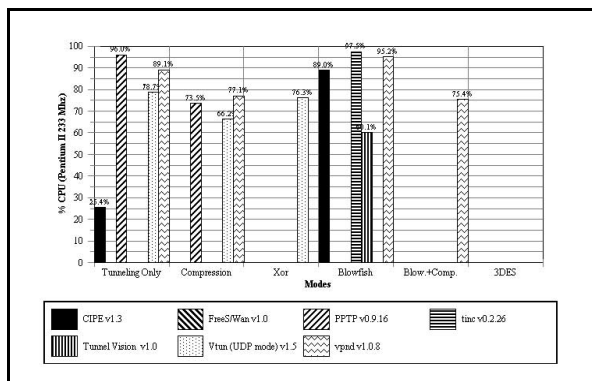


Figure 2 – CPU Usage (Server)

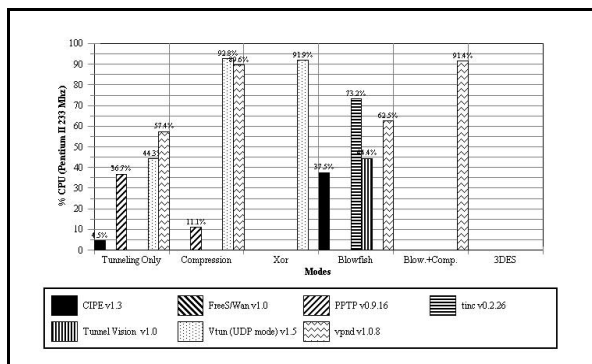


Figure 3 – CPU Usage (Client)

These CPU usage values were calculated by averaging the results provided by top with a refresh rate of 1 second (at this rate, top uses less than 10% of the CPU and affects the VPN speed by less than 2%). FreeS/Wan is a kernel process and not displayed by top.

The memory requirements are low, between 0.3% and 0.7%. While the VPN is transferring data, some processes vary their size by 0.1% or 0.2%.

5. VPN performance over a slow network

This connection was only 10.3 kb/s. The CPU usage was below 5% in all cases. FreeS/Wan was not used on the dialup host. Most VPNs showed a performance improvement due to slower transference rates and more processor availability, as displayed in Figure 5.

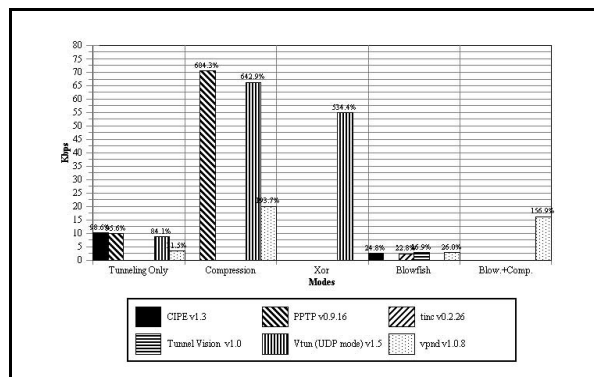


Figure 5 – Serial Throughput

6. IPsec performance of FreeS/Wan

FreeS/Wan implements the IPsec standard [1]. The difference between the different modes is 0.3% only.

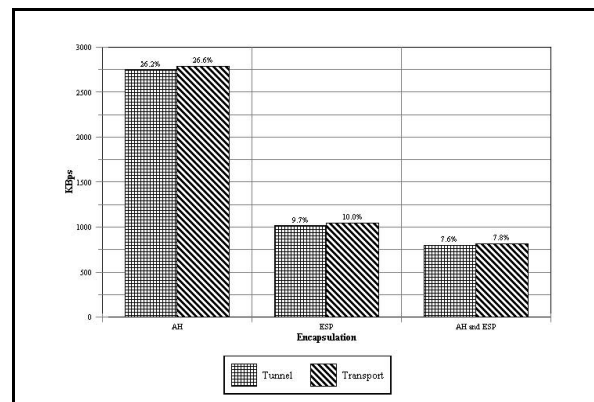


Figure 6 – FreeS/Wan Throughput

7. Conclusion

When the network connection is fast, it seems that the software solution is not efficient enough to handle the data transmission. When the network connection is slow, the CPU is not overloaded and compression could give an additional benefit depending on the data type. Code optimizations, such as the use of assembly for the encryption routines, would result in speed improvements.

Future work should focus on the testing of newer software with faster hardware to help generalizing these observations.

References

- [1] RFC 2401, Security Architecture for Internet Protocol, 1998