IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 9, Issue 6, Dec - Jan, 2022
ISSN: 2320 – 8791 (Impact Factor: 2.317)
www.ijreat.org

Optimizing VPN Speed for Remote Operation Using ARM Routers

Yasir Ali Matnee[1], Bahaa Abdul qader Thabit[2]

Department of Computer Science, Basic Education College, University of Diyala, IRAQ[1,2]

Abstract: This article is devoted to the tasks of optimizing the speed of a VPN connection when using routers with ARM processors. In modern conditions, at many enterprises and institutions around the world, there is an urgent issue of providing access for employees, as well as a remote branch or subdivision to the resources of the local network of the head office. The article discusses the possibility of connecting employees via an encrypted VPN channel using modern household routers with ARM processors. With this approach, automatic connection of all devices of a remote user to the resources of local networks of the head office and the enterprise is provided, and there is no need for IT-specialists of the enterprise to configure each device of the user separately. The paper considers the solution to the key problem of this approach, namely, ensuring the maximum speed of the encrypted VPN connection and, therefore, accelerating the software components of the routers included in its software for a high-speed encrypted VPN connection. The optimization of the speed of encryption and decryption algorithms using the features of the target processor of the device is considered, such as parallelizing the execution of processor instructions using SIMD, a general improvement in router performance when using optimal compiler options, unconventional use of PCI hardware encryption devices, the use of alternative options for modern VPN networks for routers with a relatively low-power ARM central processor in clock frequency, but containing more than two cores, while providing multithreading of the VPN channel.

Keywords: open ssl protocol, virtual private network open VPN, extension of arm instructions neon, computer calculations simd, arm architecture, routers, remote access

INTRODUCTION

The problem of providing access for employees of enterprises and institutions, as well as remote branches and divisions to the resources of the local network of the head office does not lose its relevance all over the world.

Remote users, often even in another city or country, are forced to use open Internet networks, so the question of ensuring data security and preventing information leakage arises. To solve the problem of preserving the security of data and information, if it is necessary to remotely access local network resources and databases of an enterprise / institution, an approach using virtual private networks, or VPN (Virtual Private Network), is used.

It works as follows: a virtual digital channel is created on top of open networks, and the interception of traffic by third parties will not allow them to gain access to classified information, since this uses reliable encryption algorithms for the data stream.

The VPN solution is also suitable for individuals who are often forced to use open, unencrypted networks in cafes or hotels to ensure the privacy of access to their home archives.

Using routers for VPN connections

Currently, the vast majority of VPN solutions provide cross-platform compatibility, demonstrating excellent performance on various types of devices (Windows, Mac, Linux, iOS, Android), both when using special clients and when manually configured. However, in most cases, for system administrators of head enterprises and institutions, such an approach using separate configuration files for VPN clients for each employee of the enterprise, which often has several devices (computer, phone, tablet), is unacceptable due to the laboriousness of setting up and maintaining all these devices. As a solution in such cases, the centralized use of routers (routers) is applicable, maintaining a constant connection to the head VPN server of the enterprise / institution 24 hours a day and providing simultaneous access to the data of the local network resources of the enterprise for all its users connected with a wired or wireless connection. With this approach, IT professionals and home users only need to configure and maintain the router itself, thereby automatically providing a VPN connection for various devices at the same time.

In addition, a relatively inexpensive household router can also act as a cheap home VPN microserver for the needs of an individual, ensuring the privacy of his connection when using open Wi-Fi networks. These household routers are SOHO (Small Office, Home Office), affordable, and can be used to connect a relatively small branch office or worker's home to a central VPN server.

Problem when using routers for VPN connections

Currently, the most widespread is OpenVPN [1, 2] open source, used to create encrypted channels and freely distributed under the GNU GPL [3]. The network allows you to establish a connection between computers and devices behind NAT and firewall without changing their settings. The data transmitted and received by the VPN server goes over an encrypted channel, the protection of which ensures security and privacy. Data encryption and decryption occurs both on the server and on the client machine. At the same time, OpenSSL is more often used to encrypt the channel - a full-fledged open source cryptographic library [4], widely known due to the SSL / TLS extension used, for example, in the HTTPS web protocol. It supports symmetric block cipher algorithm AES [5], which is currently one of the most widely used cryptography algorithms. Thus, hardware support for AES instructions was introduced by Intel / AMD into the x86 / x64 processor family, which are mainly used in modern powerful computers - VPN servers of enterprises and institutions.

However, the vast majority of modern routers are based on the much weaker type of ARMv7-A [6] processors, controlled by a specialized version of Linux and most often with hardware accelerated NAT

and wireless connection (firmware), but do not support AES encryption at the level of instructions from the central processor. control (CPU). Thus, when using routers, the speed of the OpenSSL library is critical. It is imperative that encryption and decryption does not slow down the speed of the OpenVPN channel, thereby providing quick access for users to the resources of the remote local network.

Methods for accelerating OpenSSL work, taking into account the specific capabilities of an ARM processor

Let's take a look at a common mistake made by router software developers. Often they are not permanent employees of the router manufacturer, but are temporarily hired in an outsourced format.

OpenSSL is a cross-platform solution, and its standard compilation by the compiler, like its use itself, is possible for various types of processors (x86 / x64, ARM, MIPS, PowerPC, etc.). However, such a formal generalized approach to compiling OpenSSL does not fully reveal all the capabilities of the target processor, which leads to degradation of the encryption and decryption speed, thereby slowing down the VPN connection channel. The first processor of the ARM architecture was created back in 1985, and since then its architecture has been constantly improved and supplemented. So, for example, ARMv7-A based on the Cortex-A9 core, which is widely used in routers, optionally supports an advanced SIMD (Single Instruction, Multiple Data, called NEON technology, which allows for parallelism of computations, and therefore .

Let's consider an example of a mistake made by the developers of a company that creates the official firmware for the R9000 router from NETGEAR. At the same time, the R9000 is positioned by NETGEAR as the fastest router in the world. It is equipped with a reasonably powerful 1.7GHz quad-core ARM processor with a Cortex-A15 core. Let's present the results of the OpenSSL tests performed on the official router firmware:

The 'numbers' are in 1000s of bytes per second processed.

type	16 bytes	64 bytes	256 bytes	1024 bytes	8192 bytes
sha1	19729.61k	54213.54k	111554.18k	150575.10k	168700.40k
des cbc	33284.58k	34141.59k	34585.00k	34665.81k	34553.86k
des ede3	12548.81k	12727.87k	12788.65k	12801.71k	12782.25k
aes-128 cbc	57205.07k	60562.69k	62545.32k	63109.12k	63310.51k
aes-192 cbc	50571.55k	52632.14k	53764.35k	54159.02k	54274.73k
aes-256 cbc	44746.83k	45857.66k	47048.96k	47419.08k	47363.41k
sha256	13311.57k	29732.76k	50673.44k	61281.28k	65227.43k

The results of the same test with specialized use of NEON instructions and some other Cortex-A15 capabilities for OpenSSL in the firmware version created by the authors of this article on the same router look like this:

The 'numbers' are in 1000s of bytes per second processed.

type	16 bytes	64 bytes	256 bytes	1024 bytes	8192 bytes
sha1	21691.86k	67717.40k	163728.90k	251297.48k	296394.75k
des cbc	33224.61k	34769.92k	35351.13k	35573.21k	35370.33k
des ede3	13231.06k	13375.81k	13498.79k	13595.49k	13485.29k
aes-128 cbc	76702.52k	80093.80k	83207.17k	84156.70k	83875.16k
aes-192 cbc	61568.46k	66469.16k	70230.95k	71435.13k	71363.24k
aes-256 cbc	55345.12k	57141.60k	58567.85k	58935.30k	59026.09k
sha256	24173.65k	56915.65k	102226.09k	128476.16k	139047.56k

As you can see, the speed of the AES algorithm has increased to 33%, the hashing algorithm SHA-512 - more than 3 times, and the RSA algorithm signature and verification - 4-5 times. This means that the speed of the OpenVPN connection for a given router can be increased by about 35-50% using the alternative optimized firmware and, therefore, the "fastest router in the world" can be significantly accelerated.

Even more interesting benchmark results can be obtained by applying the same optimizations for routers with Qualcomm's 1.4GHz dual-core IPQ8064 processor and Snapdragon Krait core. This processor is used in NETGEAR R7500, R7500v2, ZyXEL NBG6816, ASUS RT-AC87U routers and in many models from other manufacturers. So, for the AES-256-CBC algorithm in the official firmware from NETGEAR, a speed of 24443.80 k is observed for a 1 KB block, and in the firmware optimized by the authors, the result is 42048.00 k, that is, the speed of an OpenVPN connection can be, therefore, must be increased by at least 70%.

Methods for speeding up OpenSSL with non-standard use of additional hardware routers

Many modern routers, although they do not have support for AES instructions in the CPU according to the ARMv7-A specification, are equipped with special hardware cryptographic accelerators, which are most often used in the closed source driver chipset manufacturers for these routers. However, the use of such hardware accelerators is quite applicable to accelerate OpenSSL, and, consequently, to increase the speed of the OpenVPN channel, which, unfortunately, is very often ignored by the developers of official firmware. So, the same R9000 router from NETGEAR is equipped with a special PCI hardware cryptographic accelerator device, but the official firmware does not use these capabilities to accelerate OpenSSL, as well as the capabilities of the CPU itself, which is described above.

To access the capabilities of such hardware cryptographic accelerators, the kernel driver access method is usually used, since the accelerators themselves support instructions that differ from those of the CPU.

General optimization methods using the compiler's capabilities and taking into account the specifics of the target ARM processor

Another typical mistake made by the developers of the official router software is the incorrect set of options for the GNU C / C ++ compiler, which is used when building the entire firmware, and therefore

OpenVPN. Typically, this set of options is set once in the configuration file and then automatically used to build all Linux firmware and kernel software packages. This is especially true for non-standard target ARM-compatible CPUs, such as Qualcomm's Snapdragon family of Krait processors, licensed from ARM. For example, consider the 1.7GHz Qualcomm IPQ8065 dual-core processors used in routers such as NETGEAR R7800, Synology RT2600ac, ASUS BRT-AC828, and several other manufacturers' routers.

The developers of the official firmware mistakenly assume that this processor with the Krait core is a clone of the ARM Cortex-A9 core, not taking into account that, in terms of the set of instructions and supported extensions, Krait is rather a clone of the more advanced Cortex-A15 core, which has significantly higher performance and opportunities due to these extensions. Moreover, such a mistake was made at one time by the developers of the improved version of the GNU C / C ++ Linaro compiler (Linaro is a non-profit organization that consolidates and optimizes open source software for ARM platforms) and even the developers of the well-known alternative firmware OpenWRT / LEDE .

For example, the official firmware version for the NETGEAR R7800 router still uses compiler options, for example –march = armv7-a and –mfpu = vfpv3-d16, while a CPU with a Krait core supports a more advanced FPU - VFPv4, and using –march = armv7-a prevents, for example, instructions such as SDIV / UDIV for integer division, which are much faster than similar standard ARM instructions. That is, the instruction set for the Cortex-A15, and therefore for the Krait, is a superset of the ARMv7-A instructions. Thus, simply replacing these compiler options with –mcpu = cortex-a15 and –mfpu = neon-vfpv4 allows you to get a 10-15% increase in OpenVPN performance, as well as overall router performance.

For example, the test results for calculating the speed of working with RAM and calculating the numbers Pi and E using the options selected by the developers of the official firmware (–march = armv7-a and –mfpu = vfpv3-d16) on a router with an IPQ8065 processor will be as follows:

Time to run memory bench: 0.49[secs]

Time to run computation of pi (2400 digits, 10 times): 2.88[secs]

Time to run computation of e (9009 digits): 2.41[secs]

And the same test with the –mcpu = cortex-a15 and –mfpu = neon-vfpv4 options:

Time to run memory bench: 0.43[secs]

Time to run computation of pi (2400 digits, 10 times): 1.50[secs]

Time to run computation of e (9009 digits): 1.43[secs]

So, you can observe a significant increase in the speed of calculations, which leads to an increase in the speed of the OpenVPN channel (as well as to an increase in the overall performance of the router).

WireGuard Virtual Private Network

Many home SOHO routers do not have a powerful enough processor to provide high-speed connections using OpenVPN. However, a significant proportion of these routers are equipped with a quad-core ARM processor. When using OpenVPN, this multicore CPU does not matter, since OpenVPN runs in user space and uses only one thread for connections, that is, one CPU core.

However, at present, a new type of virtual private network, WireGuard, is gaining increasing popularity . It is a free, completely open source VPN. The creators position it as "extremely easy to use, the fastest VPN currently using state-of-the-art cryptography algorithms."

One of the main advantages of WireGuard is that this VPN is included in the Linux kernel (kernel space) and supports multi-threading unlike OpenVPN. Thus, the key feature of the router for working with this VPN is not the CPU clock speed, but the number of processor cores.

Thus, a relatively low-performance router of the NETGEAR Orbi RBK50 system (a tri-band mesh system for seamless Wi-Fi coverage of an extended area of a home or office) with an IPQ4019 ARM processor (ARM Cortex-A7 processor, clock frequency 710MHz) cannot provide an acceptable channel speed OpenVPN, significantly outperforming, for example, routers with an ARM processor IPQ8065. However, the IPQ4019 processor is quad-core, and when using WireGuard, the VPN speed is significantly faster than the OpenVPN speed when using the IPQ8065 processor.

Although VPN WireGuard does not use the OpenSSL library, the authors have been able to use many of the WireGuard acceleration techniques described above. So, the ARM Cortex-A7 processor also has a SIMD NEON block, that is, the encryption and decryption of the VPN channel can be parallelized not only across the processor cores, which is provided by VPN WireGuard itself (multi-threading), but also in each core (SIMD NEON).

In addition, the software developers of the official firmware of this system repeated all the same mistakes as before with other routers, not taking into account that the Cortex-A7 in the set of instructions absolutely coincides with the Cortex-A15 and with Krait, using fairly general options of the GNU compiler for the architecture ARM, such as –march = armv7-a and -mfpu = vfpv3-d16.

In addition, the developers of the official firmware made another very serious mistake when using the -mfloat-abi = soft option. In practice, this option completely excludes the FPU from work, that is, the FPU is completely turned off for floating point arithmetic, which leads to degradation of the overall performance of the router.

By replacing the options chosen by the developers of the official firmware with the optimal ones for the given target type of ARM processor, that is, -mcpu = cortex-a7, -mfpu = neon-vfpv4, -mfloat-abi = soft, it was possible to increase the overall system performance by an average of 10 -15 %. For example, a

specialized test for calculating the numbers Pi and E, compiled with the options of the developers of the official firmware, gives the following results:

Time to run memory bench: 1.94[secs]

Time to run computation of pi (2400 digits, 10 times): 3.61[secs]

Time to run computation of e (9009 digits): 4.04[secs]

And compiled with options selected by the authors:

Time to run memory bench: 1.63[secs]

Time to run computation of pi (2400 digits, 10 times): 3.12[secs]

Time to run computation of e (9009 digits): 3.38[secs]

This optimization has shown itself in real tests, showing quite surprising results. So, with the actual speed of the direct Internet connection on average 190/190 Mbps, the speed when using VPN WireGuard for the Orbi system turned out to be higher, exceeding the speed of the actual connection (198/196 Mbps).

Conclusion

The VPN speed optimization methods discussed above are included by the authors in Voxel firmware for NETGEAR R7500 / R7800 / R9000 routers, Orbi RBK50, which is used by thousands of owners of these routers around the world, successfully competing with both the official firmware from NETGEAR and other alternative firmware , and allows you to get a high speed VPN-channel.

References

1. Jose E.; Christian A.; Cristian D. Open VProxy: Low Cost Squid Proxy Based Teleworking Environment with OpenVPN Encrypted Tunnels to Provide Confidentiality, Integrity and Availability IEEE Engineering International Research Conference (EIRCON)2020.

2. OpenVPN. URL: https://openvpn.net/ (date of access: 25.12.2021).

3. GNU General Public License. URL: https://ru.wikipedia.org/wiki/GNU_General_Public_License (date accessed: 19.04.

4. J Tsai - Berkeley , For better or worse: Introducing the GNU General Public License version 3 ,Technology Law Journal, 2008 .

5. N Ferguson, B Schneier, rizkia.staff Practical cryptography, telkomuniversity.ac.id2003 - .

6. K Zhang, H Su, P Zhang, Y Dou . Optimization and Performance Modeling of Stencil Computations on ARM Architectures, IEEE International Conference on High Performance Computing and Communications (HPCC) ,2020.