Avaliação de desempenho de tecnologias VPN atuais*

Carlos Rodrigues[1,3] and Nuno Cruz[1,2][0000−0001−8570−8670]

[1] FIT - Future Internet Technologies, ISEL - Instituto Superior de Engenharia de Lisboa, Instituto Politécnico de Lisboa
[2] LASIGE, Faculdade de Ciências, Universidade de Lisboa
[3] SOLVIT - Innovation and Development on Telecomunications, Lisboa
cerodrigues@deetc.isel.ipl.pt, nuno.cruz@isel.pt

Abstract. Hoje em dia a utilização de Redes Virtuais Privadas (em inglês VPN) está cada vez mais em voga. O ritmo binário alcançado com a utilização de VPN é bastante importante na medição do desempenho das mesmas, mas não é o único critério que deve ter sido em conta. Em ambientes computacionais mais limitados ou que devem atender ao tratamento de dados de natureza crítica, a utilização de CPU é um fator de elevada importância, bem como a latência. Neste trabalho pretende-se avaliar o ritmo binário alcançado, latência, utilização de CPU e perda de pacotes para as diversas cifras suportadas pela OpenVPN, Wireguard e IPsec (strongSwan), utilizando o sistema operativo Linux. Vão ser testadas num ambiente com suporte a aceleração de hardware (AES-NI), com ligações de 1 Gbps e apenas as cifras recomendadas para cada VPN serão tidas em conta.

Keywords: VPN · OpenVPN · Wireguard · IPsec · UNIX

1 Introdução

A utilização de VPN, quer por motivos de segurança ou para estabelecer ligações entre diversas redes separadas entre si, é cada vez mais popular. Assim sendo, é de tremenda importância realizar uma avaliação de diversas soluções existentes no mercado de forma a entender qual a que se enquadra melhor nos objetivos que se pretendem atingir.

As VPN diferem entre si em parâmetros como o protocolo de transporte utilizado, cifras e mecanismos de autenticação. Estes parâmetros afetam o desempenho das VPN, seja em termos de ritmo binário, latência ou perda de pacotes.

Dependendo do caso de utilização, pode ser preferível o uso de uma VPN com a qual seja possível atingir um ritmo binário superior. Já noutros cenários,

* Este trabalho é financiado pela Agência Nacional de Inovação, através do projeto Ferrovia 4.0, ref. LISBOA-01-0247-FEDER-046111 & POCI-01-0247-FEDER-046111 and by Fundação para a Ciência e a Tecnologia, através do LASIGE, ref. UIDB/00408/2020 and ref. UIDP/00408/2020

pode ser mais relevante um valor baixo de latência, como, por exemplo, num sistema de suporte a serviços críticos.

Neste trabalho pretende-se avaliar o ritmo binário alcançado, latência, utilização de CPU e perda de pacotes para as diversas cifras suportadas pela OpenVPN, Wireguard e IPsec (strongSwan), utilizando o sistema operativo Ubuntu. Vão ser testadas num ambiente com suporte a aceleração de hardware (AES-NI) [11], com ligações de 1 Gbps e apenas as cifras recomendadas para cada VPN serão tidas em conta [2] [10].

Comparações entre OpenVPN, Wireguard e IPsec considerando o ritmo binário, latência e utilização de CPU foram realizadas também em [4]. Em [7] é também avaliado o ritmo binário, considerando variações de utilização de AES-NI, CPU turbo *boost* e CPU *pinning*. A versão de Wireguard que foi testada é anterior à sua inclusão no Kernel. Em [5] as VPN são comparadas em termos de ritmo binário, considerando variações no tamanho dos pacotes e quantidade de *flows*. É ainda avaliado o código de cada VPN de forma a entender o que limita o seu desempenho. A versão da Wireguard testada foi uma versão anterior à sua inclusão no Kernel.

Na Secção 2 deste documento é realizado um estado da arte das VPN avaliadas, enquanto na secção 3 são descritas as condições nas quais os testes foram realizados. Por fim, na Secção 4, são descritos, analisados e comparados os resultados dos testes realizados para avaliar as VPN com base nos indicadores mencionados.

2 Estado da Arte

2.1 OpenVPN

A OpenVPN [1] é uma VPN *open source* que permite a criação de canais seguros entre *endpoints*, através de *Transport Layer Security* (TLS). É uma das VPN mais populares dada a sua facilidade de configuração face a outras concorrentes. Para além disso é possível utilizá-la em praticamente qualquer dispositivo, incluindo dispositivos Android e IOS. Utiliza a biblioteca OpenSSL para encriptação e autenticação, suportando todas as cifras que são suportadas pela biblioteca. Por omissão, desde a versão 2.5 da OpenVPN, é utilizada a cifra AES-256-GCM, uma cifra autenticada e que fornece ainda integridade, através da função GHASH, eliminando a necessidade de um algoritmo adicional para esse fim. Para além de AES-256-GCM, é recomendada a utilização de AES-128-GCM. A cifra ChaCha20Poly1305 é também reconhecida como uma alternativa viável [2]. Em versões anteriores a cifra utilizada por definição era AES-256-CBC e era recomendada também a cifra AES-128-CBC.

Contrariamente à maioria das VPN, em que apenas é suportado o estabelecimento de túneis através de UDP, a OpenVPN também permite a utilização de TCP. Cada um destes protocolos tem as suas vantagens e casos de utilização. Por um lado, um túnel UDP apresenta uma maior ritmo binário e uma latência menor. Por outro, é menos estável que um túnel TCP [3].

Outro aspeto importante é que pode atuar tanto no nível 2 como no nível 3 da camada OSI. No nível 2 é utilizada para criar uma ponte entre redes, enquanto no nível 3 é utilizada para criar um túnel de rede.

Um dos problemas apontados a esta VPN é que o tempo necessário para início de ligação é bastante elevado se comparado com outras VPN [4]. Outro problema é o facto de ser *single-threaded*, não sendo muito escalável em ambientes compostos por diversos CPU [5].

2.2 Wireguard

A Wireguard, segundo descrição do próprio autor e criador Jason Donenfeld [8], define-se como uma implementação moderna, rápida e segura de uma VPN, pretendendo ser mais simples, rápida e de mais fácil utilização do que IPsec. Assume-se ainda como consideravelmente mais rápida do que a OpenVPN.

Este protocolo foi desenvolvido de raiz, procurando utilizar as cifras mais atuais e seguras, atuando no nível 3 do modelo OSI. Tem vindo a ganhar notoriedade principalmente desde o momento em que foi incluído no Linux Kernel 5.6 e também pelo desempenho anunciado, que é bastante promissora. Porém, o autor reconhece que os *benchmarks* apresentados [8] não estão atualizados e não foram muito bem conduzidos, apresentando-se demasiado otimistas.

Como protocolo de transporte, apenas é possível a utilização de UDP. Ao nível das cifras, é utilizado ChaCha20 para encriptar os dados, combinada com Poly1305 para garantir autenticação, não existindo quaisquer outras opções. Desta forma, o código da implementação permanece curto e mitigam-se potenciais erros de configuração, que poderiam levar ao estabelecimento de ligações não seguras. Não são usados quaisquer certificados para o estabelecimento de ligações, evitando a necessidade da sua verificação, aumentando o desempenho da VPN.

Outro aspeto importante é que esta VPN suporta *multi thread*, sendo escalável em ambientes compostos por múltiplos CPU. No entanto, em [5] é demonstrado que, para redes de elevado ritmo binário, esta não consegue escalar de forma eficaz.

2.3 *Internet Protocol Security*

Internet Protocol Security (IPsec) é um grupo de protocolos utilizados em conjunto para adicionar uma camada de segurança ao protocolo IP, dando-se a encriptação dos dados de uma conexão entre dois ou mais dispositivos.

IPsec garante segurança dos dados tendo mecanismos de autenticação, verificação de integridade dos pacotes e encapsulamento dos mesmos num túnel seguro. Tratando-se de um grupo de protocolos normalizado, é possível estabelecer um túnel VPN entre equipamentos de diferentes marcas. Contrariamente à OpenVPN e Wireguard, IPsec atua diretamente sobre a camada IP e não sobre a camada de transporte.

Um dos grandes problemas apontados a este protocolo é a dificuldade de configuração. Ainda assim, é bastante adotado, praticamente por todos os tipos de sistemas e vendedores.

Uma das VPN mais conhecidas que utiliza IPsec é a strongSwan [9] e, dado que IPsec é considerado um protocolo muito escalável, são suportadas diversas cifras, tais como *Advanced Encryption Standard* (AES) com modo *Cypher Block Chaining* (CBC) e AES com modo GCM. strongSwan também possui suporte a *multi thread*. No entanto, a implementação do IPsec no Kernel Linux não.

2.4 Resumo de características das VPN

De forma a simplificar a comparação entre as VPN abordadas, na tabela 1 são apresentadas as características relevantes de cada implementação e na tabela 2 são apresentadas os algoritmos de cifra e integridade recomendados e testados para cada VPN.

Table 1. Características das VPN

—	OpenVPN	strongSwan (IPsec)	Wireguard
Troca de Chaves	SSL/TLS	IKEv1/IKEv2	Curve22519
Transporte	TCP/UDP	UDP	UDP
Espaço de Execução	*User Space*	*Kernel Space*	*Kernel Space*
Suporte de *Multithread*	Não	Sim	Sim
Linguagem de Programação	C	C	C

Table 2. Algoritmos de cifra e integridade das VPN.

VPN	Cifra	Integridade
OpenVPN	ChaCha20	Poly1305
	AES-CBC-256	SHA256
	AES-CBC-128	SHA256
	AES-GCM-256	GHASH
	AES-GCM-128	GHASH
strongSwan	ChaCha20	Poly1305
	AES-CBC-256	SHA256
	AES-CBC-128	SHA256
	AES-GCM-256	GHASH
	AES-GCM-128	GHASH
Wireguard	ChaCha20	Poly1305

Algo que pode impactar o desempenho de cada VPN é o espaço de execução. Para as VPN que correm em *Kernel Space*, espera-se que apresentem um desempenho melhor. A distinção entre *Kernel Space* e *User Space* prende-se, em parte,

com o acesso ao espaço de memória. No *Kernel Space* os programas têm acesso a todo o espaço de memória disponível. Por outro lado, os programas que correm no *User Space* têm acesso limitado à memória do sistema. Outra diferença é que os programas que correm em *User Space* têm acesso limitado ao Kernel, através de uma interface (*System Calls*), enquanto que os de *Kernel Space* correm em modo Kernel. Tendo em conta que os programas dependem de determinados processos que são executados no Kernel, os programas que atuam no *User Space* gastam algum tempo de execução a realizar a comunicação entre o *User Space* e o *Kernel Space*, enquanto que os programas que correm em modo Kernel não gastam tempo nestas transições.

Outro aspeto importante no desempenho das VPN recai sobre a escolha da cifra. Cifras modernas *Authenticated Encryption with Associated Data* (AEAD) como AES-GCM e ChaCha20 apresentam um desempenho melhorado face a cifras clássicas *encrypt-then-mac*. As cifras AEAD caracterizam-se por disponibilizarem tanto privacidade como integridade de forma simultânea. Desta forma, evita-se a utilização de um algoritmo à parte para o efeito, isto é, não existe necessidade de calcular um *hash* dos dados após estes terem sido encriptados. Esta vantagem reflete-se num desempenho superior face a cifras que precisam de calcular o *hash* após a encriptação dos dados, tal como as cifras que utilizam o algoritmo SHA.

3 Condições de Teste

O que se pretende avaliar com a realização dos testes é o desempenho das diversas VPN e respetivas cifras em termos de ritmo binário alcançado, utilização de CPU, perda de pacotes e latência. Considerou-se como perda de pacotes as medidas de pacotes perdidas reportadas pelo iperf3, valores estes que não consideram a diferença entre o que é pedido para ser injetado e aquilo que efetivamente é transmitido.

Através das métricas avaliadas consegue-se aferir a viabilidade de utilização de cada VPN, dependendo do ambiente e caso de utilização que se pretende.

A strongSwan e Wireguard apenas suportam túneis estabelecidos sobre UDP, enquanto a OpenVPN suporta tanto sobre UDP como TCP. Assim sendo, para esta VPN, serão testados os dois cenários.

Como topologia de rede foram utilizadas duas máquinas virtuais, instaladas em servidores físicos diferentes. A plataforma de virtualização utilizada foi a oVirt. Os dois servidores estão conectados a um *switch*, com ligações físicas de 1 GbE. Estas ligações são utilizadas pelas máquinas virtuais para comunicarem. A topologia descrita é visível na figura 1.

Na tabela 3 são apresentadas as características de *hardware* e *software*, bem como a informação das versões das VPN utilizadas.

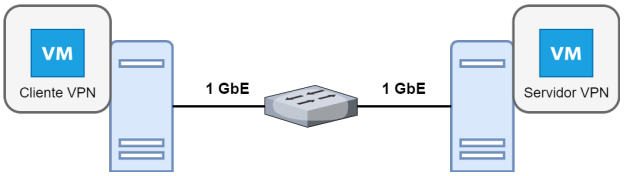

Fig. 1. Topologia de rede utilizada nos testes.

Table 3. Características de *hardware* e *software*

Sistema Operativo	Ubuntu Server 22.04 LTS
Kernel	5.15.0-39-generic
Processador	Intel Xeon E5-2609 v4
Memória	1 GB
Plataforma de Virtualização	oVirt
OpenVPN	2.5.6
strongSwan	5.9.5
Wireguard	1.0.20210914
OpenSSL	3.0.2

Em termos de ritmo binário, foram realizados 3 testes de 60 segundos para cada cifra de cada VPN, recorrendo à ferramenta iperf3 e sendo retiradas amostras a cada segundo. Para além disso, para cada VPN e cifra, foram realizados testes iperf3 UDP e TCP. No caso dos testes UDP em que o iperf3 requer como critério a largura de banda disponível, foi utilizado um valor de 1000 Mbps, o valor físico da interface de rede utilizada.

Ao nível dos testes de latência, foi utilizada a ferramenta ping, sendo enviados 50 000 pacotes ICMP a um ritmo de 1 000 por segundo.

A medição da utilização de CPU foi realizada em simultâneo com os testes de ritmo binário, a partir dos quais também foram retirados os valores da perda de pacotes. Os valores de perda de pacotes foram retirados apenas para os testes UDP, dado que o iperf3 não disponibiliza estes dados quando utilizados pacotes TCP.

4 Testes

De forma a ter uma base de comparação, foram realizados testes sem qualquer VPN, utilizando tráfego UDP e tráfego TCP (testes *Baseline*).

Os resultados apresentados foram agrupados pelo protocolo utilizado no envio dos pacotes. Em primeiro lugar serão abordados todos os resultados retirados dos testes iperf e, seguidamente, são apresentados os resultados dos testes de latência realizados com o ping.

Dado que a OpenVPN suporta a utilização de UDP e TCP como protocolo de transporte, para esta VPN, todas as cifras foram testadas a correr sobre estes dois protocolos.

4.1 Ritmo binário

Na Figura 2 são apresentados os valores de ritmo binário alcançados utilizando cada VPN, em Mbps, para tráfego UDP. Em primeiro lugar, é evidente que a OpenVPN, tanto a utilizar UDP como TCP como transporte para o tráfego da VPN é a que apresenta os valores mais baixos de ritmo binário. Em segundo lugar, verifica-se que cifras AES-GCM e ChaCha20Poly1305 conseguem melhores resultados do que as cifras AES-CBC.

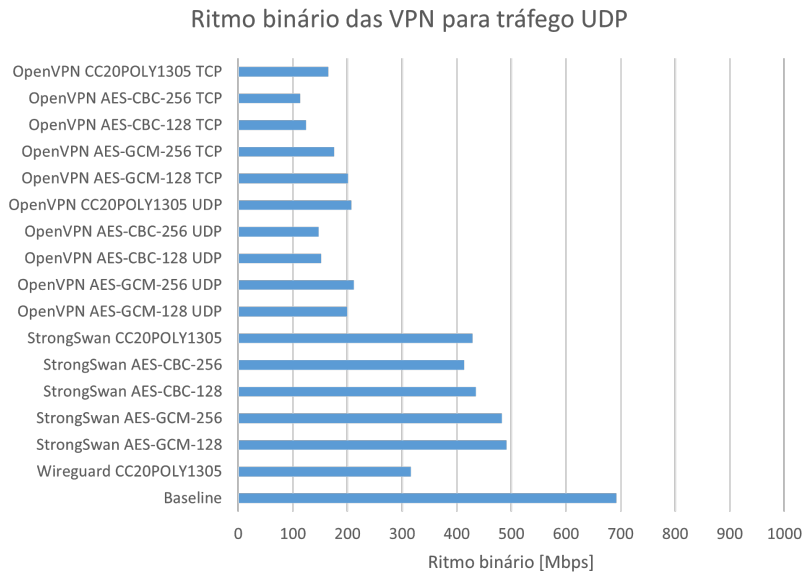

Fig. 2. Ritmo binário das VPN para tráfego UDP.

Na Figura 3 são apresentados os valores de ritmo binário alcançados para cada VPN, em Mbps, para tráfego TCP. A Wireguard apresenta um ritmo binário consideravelmente superior às restantes, conseguindo cerca de 300 Mbps a mais do que a cifra com melhor desempenho da strongSwan.

Comparando os resultados do tráfego UDP e TCP, o melhor resultado foi obtido para tráfego TCP utilizando a Wireguard que, para tráfego UDP, ficou abaixo da strongSwan com qualquer uma das suas cifras. Por outro lado, a OpenVPN foi consistente, sendo a pior em ambos os cenários. Em ambos os cenários, foi também notória a superioridade das cifras AES-GCM e ChaCha20Poly1305.

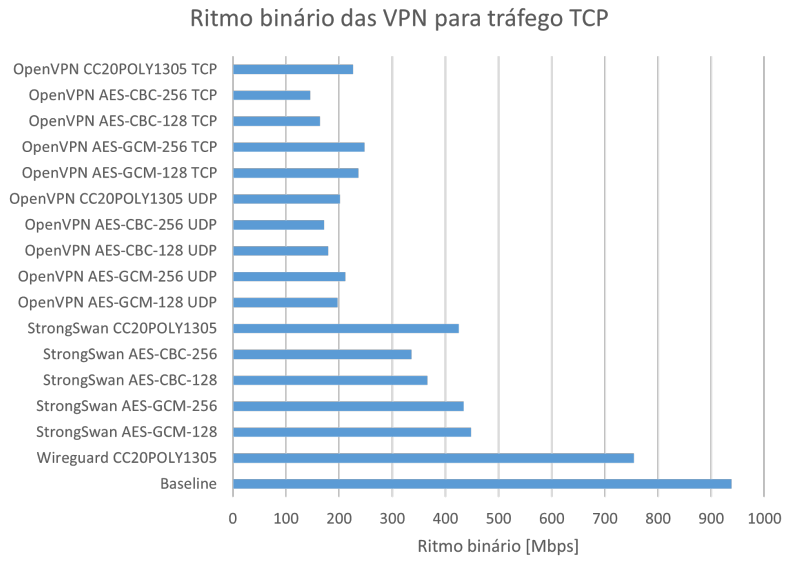

Fig. 3. Ritmo binário das VPN para tráfego TCP.

4.2 Utilização de CPU

Na Figura 4 são apresentados os valores de utilização de CPU para cada VPN, em percentagem, para tráfego UDP. De notar, desde logo, que não é apresentado qualquer valor para VPN strongswan porque não foi possível detetar qualquer processo a correr no sistema associada à mesma. Mesmo no caso deste existir, não foi observável qualquer processo a correr no sistema que apresentasse sequer um valor significativo de utilização de CPU. Para os testes *Baseline* também não existe qualquer processo, dado não ser usada qualquer VPN.

A OpenVPN apresenta valores de utilização de CPU muito similares entre si. Estes são consideravelmente elevados, principalmente considerando a utilização de CPU não significativa por parte do IPsec strongSwan. A Wireguard apresenta sensivelmente metade da utilização de CPU da OpenVPN, que, em ambientes mais limitados ou de natureza crítica, é um fator de elevada importância.

Na Figura 5 são apresentados os valores de utilização de CPU para cada VPN, em percentagem, para tráfego TCP. Tal como para o tráfego UDP, não são apresentados quaisquer valores para strongSwan e *Baseline*.

A OpenVPN, novamente, é que apresenta a maior utilização de CPU. Neste cenário, os valores são quase o dobro dos apresentados para UDP. O mesmo acontece para a Wireguard. Fica assim claro que o processamento de pacotes TCP é mais pesado para as VPN do que o processamento de tráfego UDP. Dependendo do caso de utilização, se possível, a utilização de tráfego UDP é

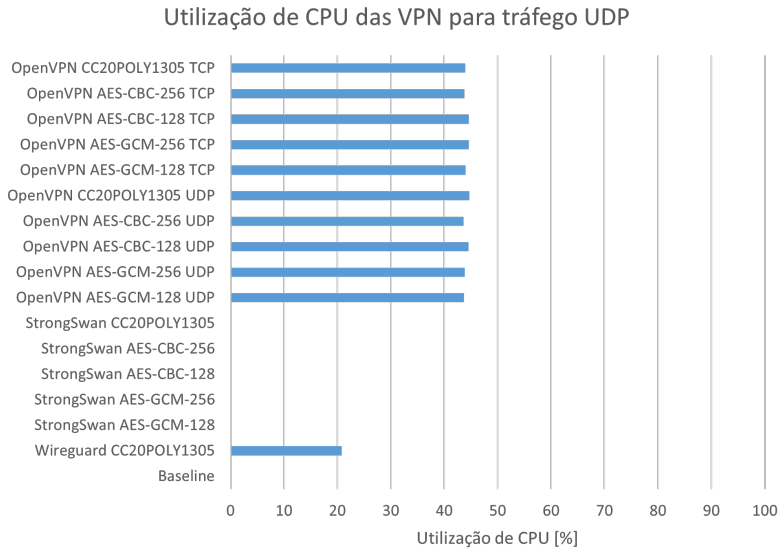

Fig. 4. Utilização de CPU das VPN para tráfego UDP.

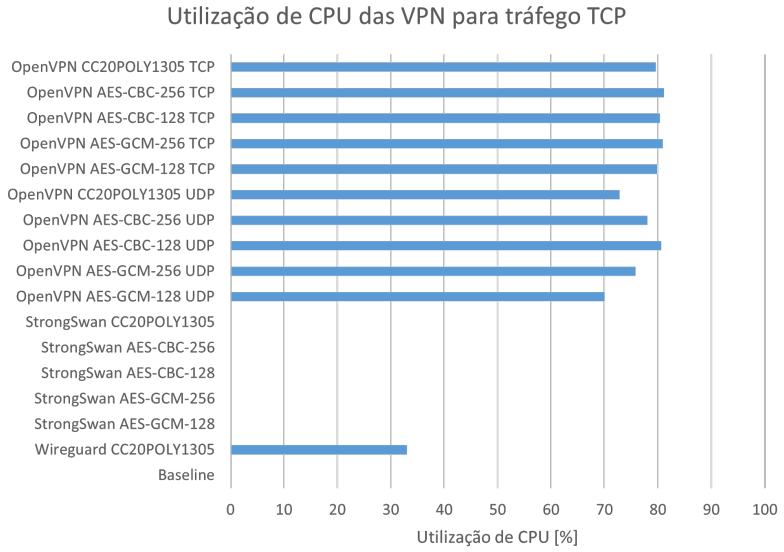

Fig. 5. Utilização de CPU das VPN para tráfego TCP.

mais vantajosa para o CPU, levando a que seja possível correr mais processos sem que exista uma sobrecarga, como é provável com tráfego TCP.

4.3 Perda de pacotes para tráfego UDP

Na Figura 6 são apresentados os valores da perda de pacotes para cada VPN, em termos de percentagem da totalidade dos pacotes enviados, para tráfego UDP.

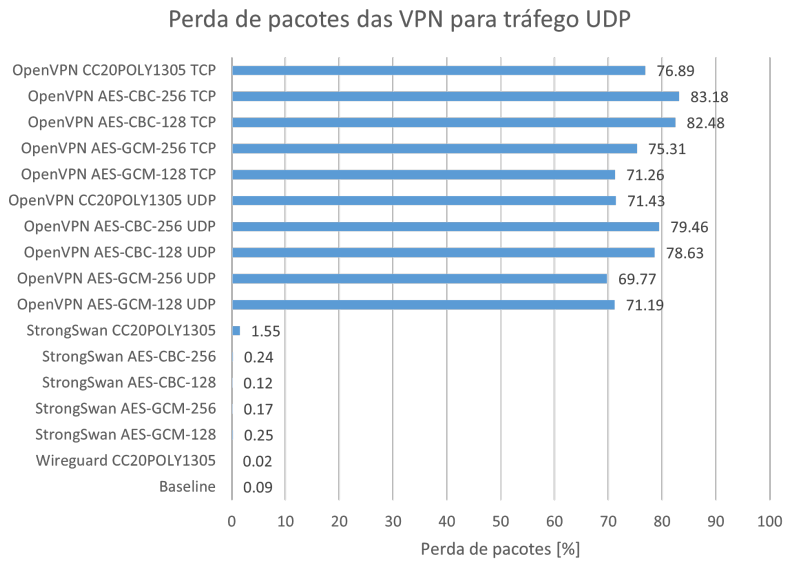

Fig. 6. Perda de pacotes das VPN para tráfego UDP.

Se por um lado a perda de pacotes para as VPN strongSwan e Wireguard é praticamente inexistente, a perda de pacotes para todas as combinações de cifra e protocolo de transporte da OpenVPN é colossal. Resultados bastante elevados de perda de pacotes foram também observados em [6]. Em todos os casos à exceção da cifra AES-GCM-128 sobre UDP, existe uma perda superior a 70%. Estes valores tão elevados levam a que a escolha desta VPN não seja viável em grande parte dos cenários. A escolha torna-se facilitada observando os resultados das restantes VPN.

4.4 Latência

Por último, foram realizados testes de latência e, na Figura 7, são apresentados os valores do RTT médio, em milissegundos, para todas as VPN. A latência da

strongSwan e Wireguard é pouco superior aos resultados *Baseline*, algo bastante positivo.

A OpenVPN apresenta os maiores valores de latência, demonstrando mais uma vez que é a VPN com pior desempenho. Para esta VPN nota-se bastante a influência da escolha do protocolo de transporte a ser usado. Em todos os casos, quando comparada a mesma cifra, sobre UDP e TCP, o melhor desempenho é obtido com a utilização de UDP. O facto da OpenVPN correr em *User Space* acaba por ter um papel fundamental nos valores obtidos, dado que introduz atrasos que acabam por impactar os mesmos. Comparações entre a latência obtida sobre túneis UDP e TCP foram também realizadas em [3], onde as conclusões são similares.

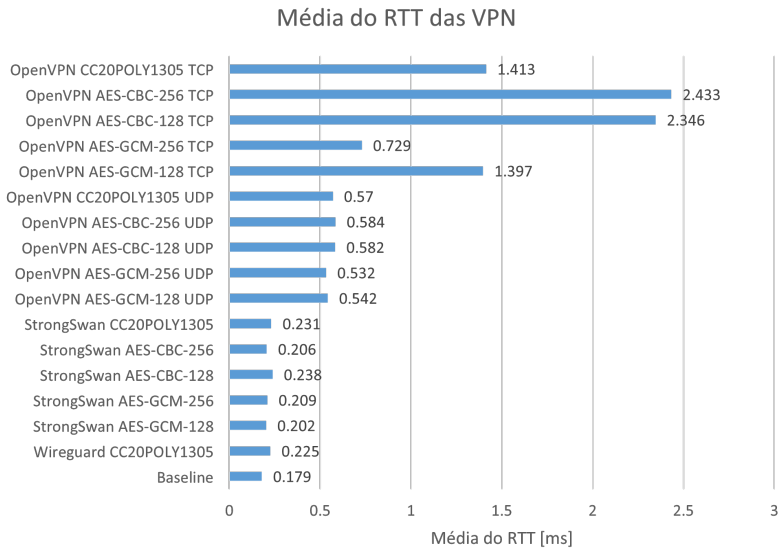

Fig. 7. RTT médio em milisegundos das VPN.

5 Conclusão

Neste artigo foi avaliado o desempenho das VPN OpenVPN, Wireguard e IPsec em termos de ritmo binário, utilização de CPU, latência e perda de pacotes. Os resultados obtidos demonstram que a OpenVPN apresenta os piores resultados em todos os testes, sem exceção. A sua utilização de CPU é muito elevada, e o mesmo acontece com a perda de pacotes. A latência é também, de longe, a mais elevada de entre as VPN testadas. No que toca ao ritmo binário, a strongSwan

apresenta os melhores valores para tráfego UDP, já em relação a TCP a Wireguard revela resultados substancialmente superiores. Em termos de perda de pacotes, existe praticamente um empate entre Wireguard e strongSwan (IPsec). Por fim, como afirmando anteriormente, não foi possível observar processos associados a IPsec com utilização significativa de CPU, pelo que consideramos que o processamento introduzido é diminuto.

O desempenho da OpenVPN ser, em todos os testes, pior do que as restantes VPN testadas era algo esperado, pelo facto desta correr no *User Space*. Desta forma, é gasto algum tempo de execução na comunicação entre o *User Space* e o *Kernel Space*, o que não acontece para a strongSwan e Wireguard.

Outro aspeto importante que se confirmou foi a superioridade geral das cifras AEAD. Estas apresentaram uma performance superior às cifras AES-CBC em todos os testes. Assim sendo, quando as cifras AEAD estiverem disponíveis, devem ser escolhidas em detrimento das cifras clássicas.

Em suma, apesar de tanto a Wireguard como a strongSwan terem pontos de destaque, revelam-se como sendo VPN bastante equilibradas e adequadas à maioria dos cenários onde se pretenda ou seja necessária a utilização de uma VPN.

References

1. OpenVPN, `https://openvpn.net/`.
2. OpenVPN Cipher Negotiation, `https://community.openvpn.net/openvpn/wiki/CipherNegotiation`.
3. I. Coonjah, P. C. Catherine and K. M. S. Soyjaudah, "Experimental performance comparison between TCP vs UDP tunnel using OpenVPN," 2015 International Conference on Computing, Communication and Security (ICCCS), 2015, pp. 1-5, https://doi.org/10.1109/CCCS.2015.7374133.
4. D. Ekker, S. Patrick, "Performance comparison of VPN implementations WireGuard, strongSwan, and OpenVPN in a 1 Gbit/s environment", 2020.
5. M. Pudelko, P. Emmerich, S. Gallenmüller and G. Carle, "Performance Analysis of VPN Gateways," 2020 IFIP Networking Conference (Networking), 2020, pp. 325-333.
6. S. T. Aung and T. Thein, "Comparative Analysis of Site-to-Site Layer 2 Virtual Private Networks," 2020 IEEE Conference on Computer Applications(ICCA), 2020, pp. 1-5, https://doi.org/ 10.1109/ICCA49400.2020.9022848.
7. Osswald, L., Haeberle, M., & Menth, M., "Performance Comparison of VPN Solutions", 2020.
8. D. Jason, "WireGuard: Next Generation Kernel Network Tunnel", 2017, https://doi.org/10.14722/ndss.2017.23160.
9. strongSwan, 2022-02-20. `https://www.strongswan.org`.
10. strongSwan Security Recommendations, `https://docs.strongswan.org/docs/5.9/howtos/securityRecommendations.html`.
11. S. Gueron, "Intel's New AES Instructions for Enhanced Performance and Security," in Fast Software Encryption - FSE 2009, ser. Lecture Notes in Computer Science 5665. Springer Verlag, 2009, pp. 51–66.