A performance comparison of the VPN implementations WireGuard, strongSwan and OpenVPN in a one Gbit/s environment

By Erik Dekker & Patrick Spaans

Supervisors: Aristide Bouix and Mohammad Al Najar

Introduction

- Organization host internal services for customers and employees.
- These often need to be reached over the internet → VPN
- Well known VPN implementations include strongSwan (IPsec) and OpenVPN
    - Often acknowledged as complex
    - Support obsolete options

[2]https://medium.com/@novysf/wireguard-server-client-with-roaming-ip-setup-498d708ebb7c

Introduction

- WireGuard!
- Aims to be simpler, faster and leaner than IPsec [1]
- Better performing than TLS based VPN solutions such as OpenVPN [1]
- Less than 4000 lines of code

[3]: www.wireguard.com/talks/lpc2018-wireguard-slides.pdf

[1]: https://www.wireguard.com/

[2]https://medium.com/@novysf/wireguard-server-client-with-roaming-ip-setup-498d708ebb7c

Introduction

- Only one cipher suite
- Fast connection setup
- Exists as a **kernel** and **Go** implementation

WIREGUARD
FAST, MODERN, SECURE VPN TUNNEL

[1]: https://www.wireguard.com/

Related work

- In 2018, Pudelko created his own VPN solutions. Additionally, he compared this with IPsec, OpenVPN and WireGuard.
- In 2020, Mackey et al. compared OpenVPN to WireGuard.
- In 2020, Osswald et al. compared IPsec, OpenVPN and WireGuard.

Gap with existing literature

- WireGuard was not implemented in the kernel yet.
- GCM ciphers for OpenVPN and IPsec were not analysed.
- Mackey et al. and Osswald et al. did not mention any configuration parameters.
- Latency was not researched before.

Main research question

How do the VPN implementations WireGuard-C, WireGuard-Go, strongSwan and OpenVPN compare in terms of performance in a 1 Gbit/s environment?

Research questions

How do the VPN implementations compare in terms of:

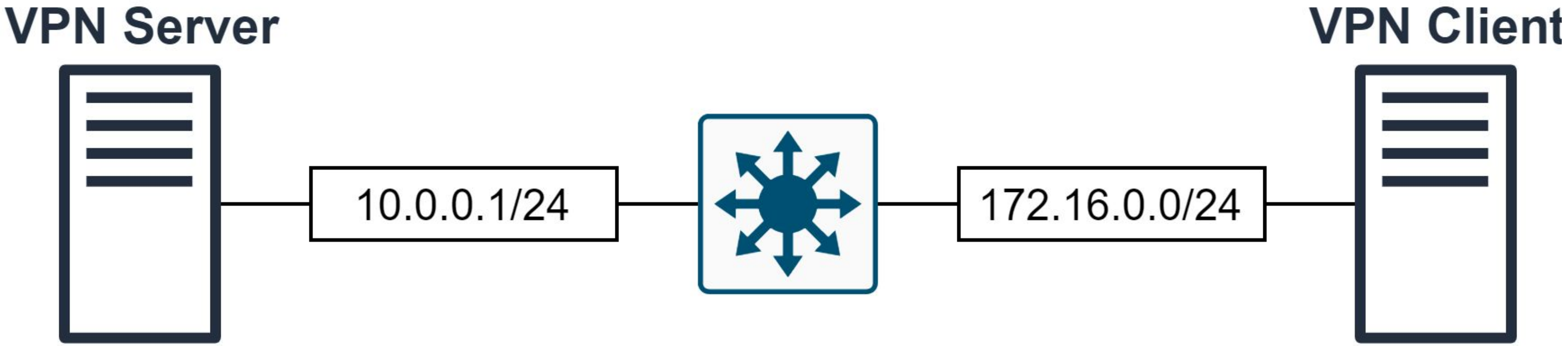
- TCP goodput
- UDP goodput
- Latency
- Connection initiation time
- CPU efficiency

Main differences

	strongSwan	OpenVPN	WireGuard-C	WireGuard-Go
Multi-threaded	Yes*	No	Yes	Yes
Key exchange	IKEv1/IKEv2	SSL/TLS**	WG	WG
Cipher	Configurable	Configurable	ChaCha20	ChaCha20
Integrity	Configurable	Configurable	Poly1305	Poly1305
User/Kernel space	Kernel	User	Kernel	User
Language	C	C	C	Go

*The current kernel IPsec is not multithreading capable
**Has it own implementation of TLS

Methodology - lab setup



Methodology - VPN configurations

Only researched the recommend cipher suites

VPN Solution	Encryption	Integrity
strongSwan	AES-128-CBC AES-128-GCM AES-256-GCM ChaCha20	SHA256 GHASH GHASH Poly1305
OpenVPN	AES-128-CBC AES-128-GCM AES-256-CBC AES-256-GCM	SHA256 GHASH SHA256 GHASH
WireGuard-C	ChaCha20	Poly1305
WireGuard-Go	ChaCha20	Poly1305

Methodology - goodput and CPU efficiency

Created a test setup and:

- Used iPerf to measure goodput.
- Used packet sizes of 64, 256, 512, 1024 and maximum bytes. As is recommended by RFC 2544.
- Calculated the most ideal packet lengths for each VPN implementation.
- Whilst doing the goodput measurements, we measured the CPU initialization with the tool mpstat.

VPN Solution	Encryption	UDP payload	TCP payload
strongSwan	AES-CBC	1410	1386
strongSwan	AES-GCM	1418	1394
strongSwan	ChaCha20	1418	1394
OpenVPN	AES-CBC	1375	1351
OpenVPN	AES-GCM	1420	1396
WireGuard	ChaCha20	1392	1368
Baseline	ChaCha20	1472	1448

Methodology - latency

- For each cipher suite we had send one million ICMP echo requests.
- Interval of 1000 per second.

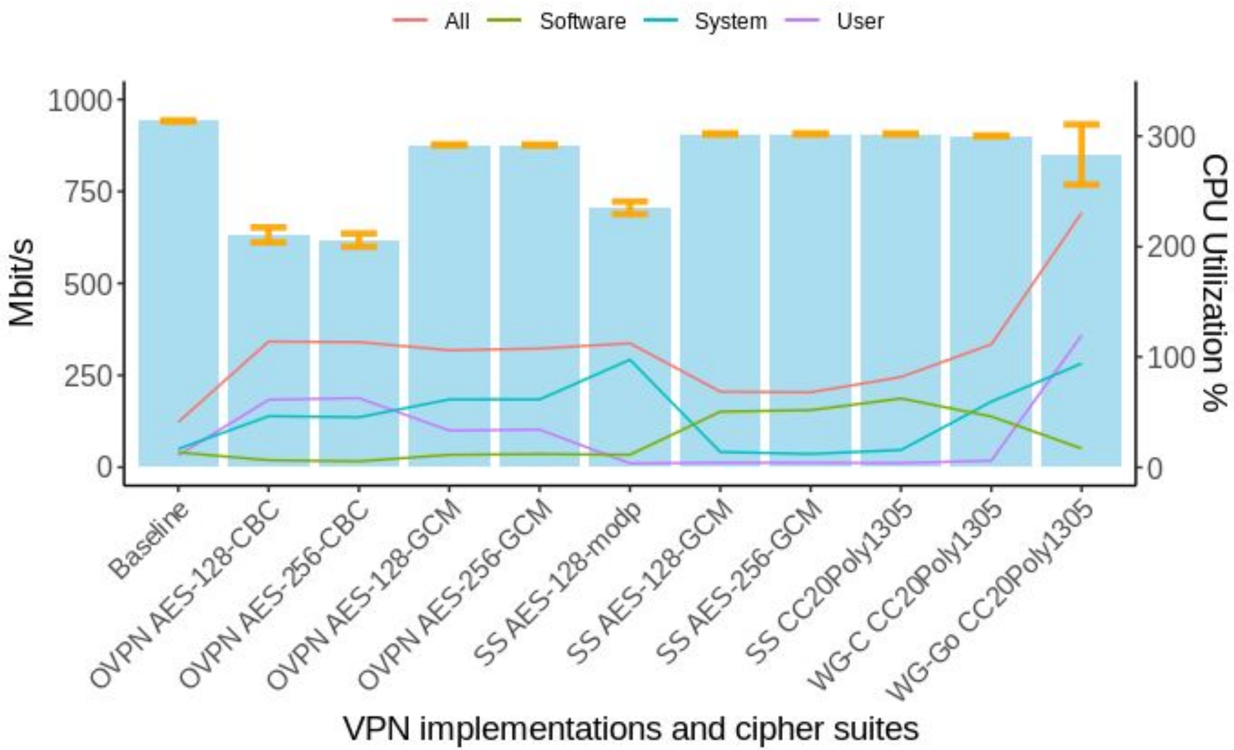
Methodology - connection initiation time

- We calculated the connection initiation time (x1000).
- We wrote a python script that looked for log messages and calculated the time difference from startup.
- We measured the time difference between the first and last connection initiation packet.

Results

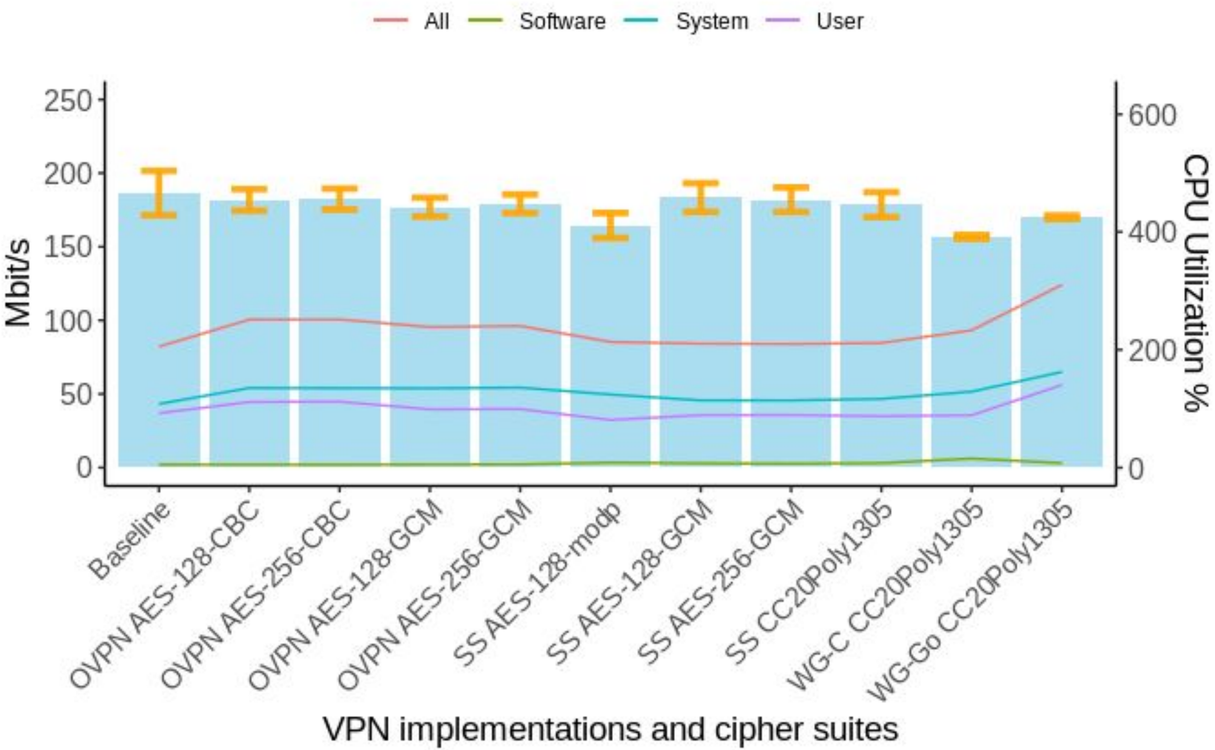
- TCP Goodput and CPU utilization
- UDP Goodput and CPU utilization
- Latency
- Initiation Time

Results - TCP & maximum packet size



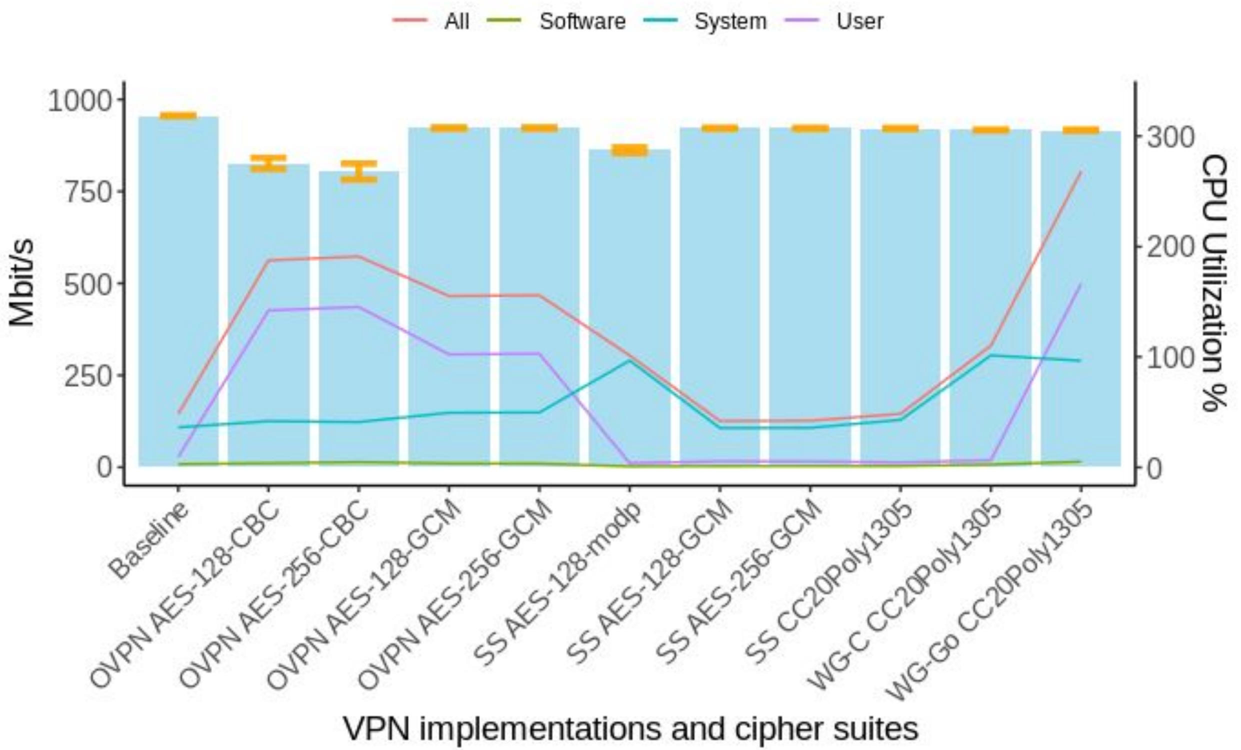
Implementation	Mbit/s
Baseline	941
OVPN AES-256-GCM	876
SS AES-256-GCM	906
WG-C CC20Poly1305	901
WG-Go CC20Poly1305	850

Results - TCP & packets of 64 bytes



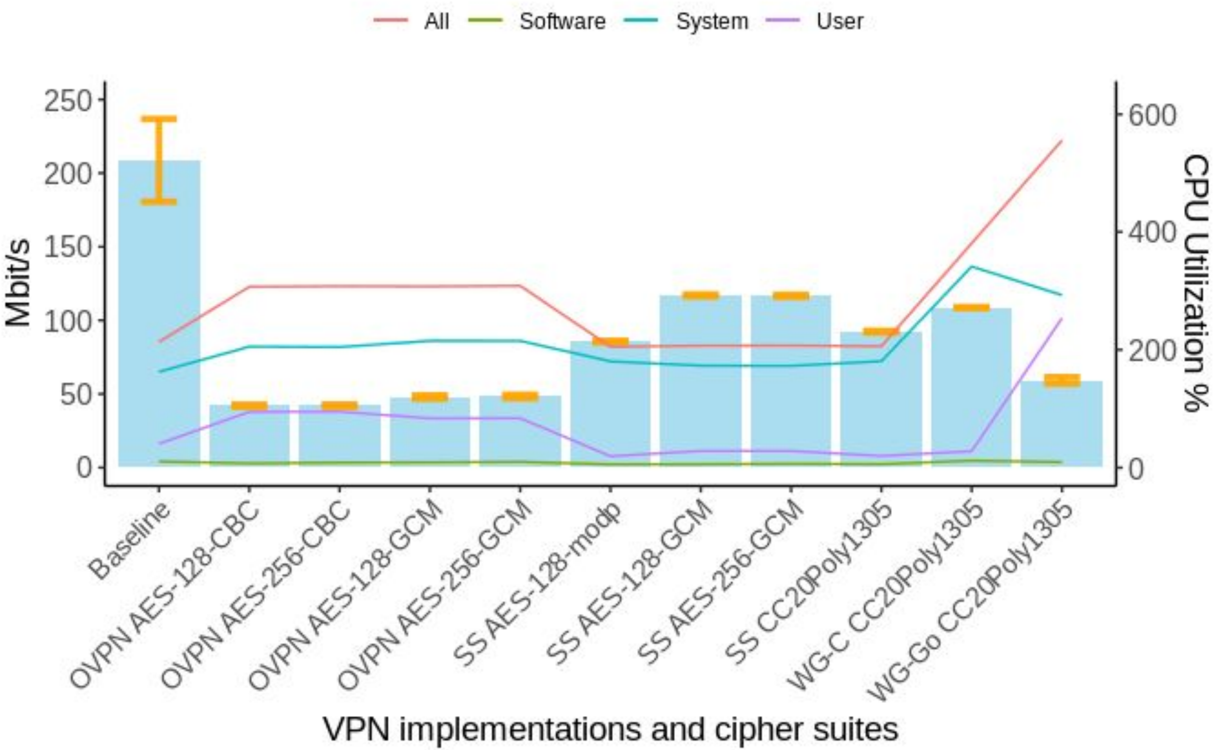
Implementation	Mbit/s
Baseline	186
OVPN AES-256-GCM	179
SS AES-256-GCM	178
WG-C CC20Poly1305	156
WG-Go CC20Poly1305	170

Results - UDP & maximum packet size



Implementation	Mbit/s
Baseline	955
OVPN AES-256-GCM	922
SS AES-256-GCM	921
WG-C CC20Poly1305	917
WG-Go CC20Poly1305	916

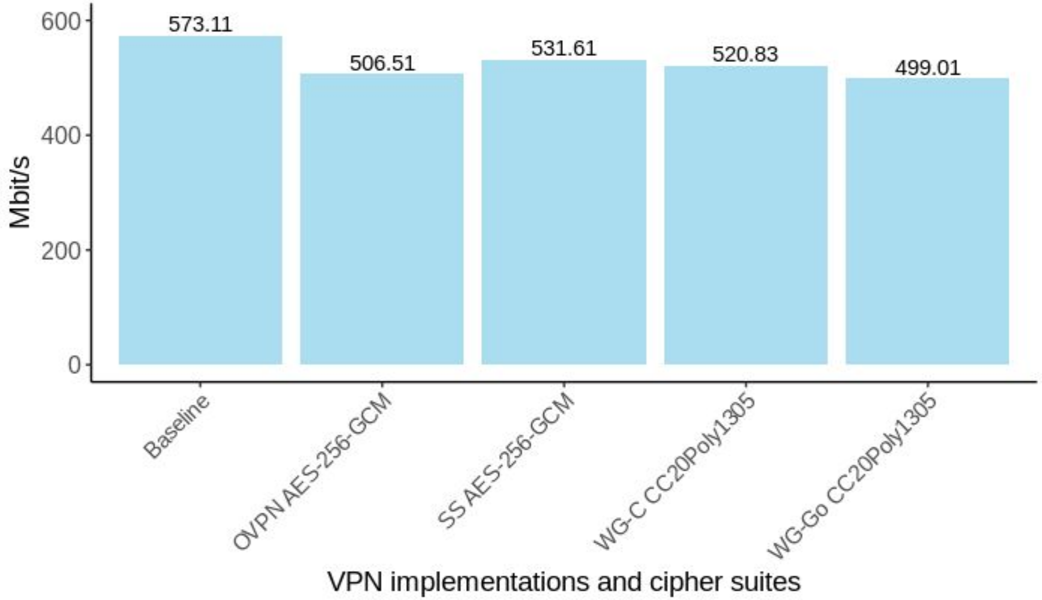
Results - UDP & packets of 64 bytes



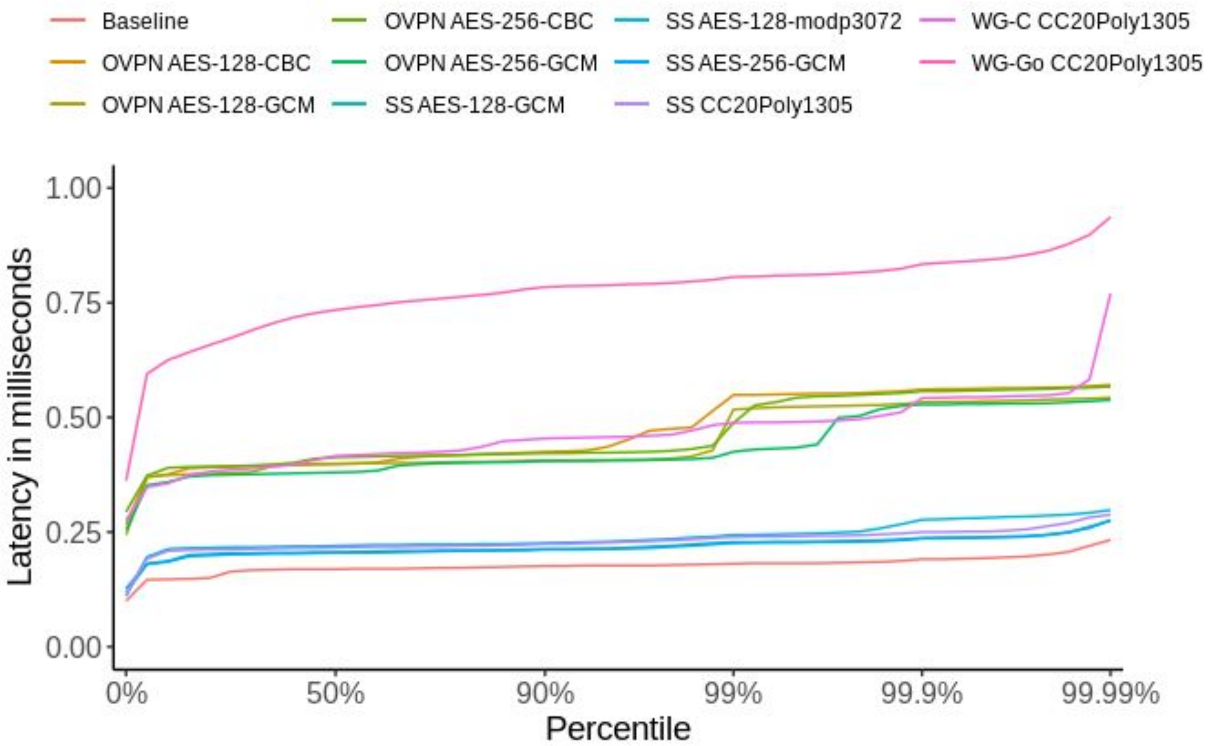
Implementation	Mbit/s
Baseline	209
OVPN AES-256-GCM	48
SS AES-256-GCM	117
WG-C CC20Poly1305	109
WG-Go CC20Poly1305	59

Summary - goodput and CPU utilization

- strongSwan AES128 GCM, AES256GCM and Chacha20Poly1305 consistently among the best.
- OpenVPN AES128 GCM and AES256 GCM perform quite well, and are only slightly behind strongSwan in terms of goodput and utilization.
- WireGuard-C generally performs slightly worse than the three strongSwan ciphersuites.
- WireGuard-Go has high CPU usage without reaching as great of a goodput.



Results - latency



Results - connection initiation time

VPN	Average	50%	90%	99%
OpenVPN (Total)	1153.7	1151.8	1261.4	1285.5
OpenVPN (Handshake)	1144.9	1144.9	1254.4	1279.1
strongSwan (Total)	33.6	33.7	34.6	35.5
strongSwan (Handshake)	4.6	4.6	4.9	5.1
WireGuard-C (Total)	6.9	7.8	7.9	8.0
WireGuard-C (Handshake)	0.7	0.7	0.8	1.1
WireGuard-Go (Total)	10.6	10.6	10.7	10.9
WireGuard-Go (Handshake)	1.0	1.0	1.1	1.1

Initiation time shown in milliseconds

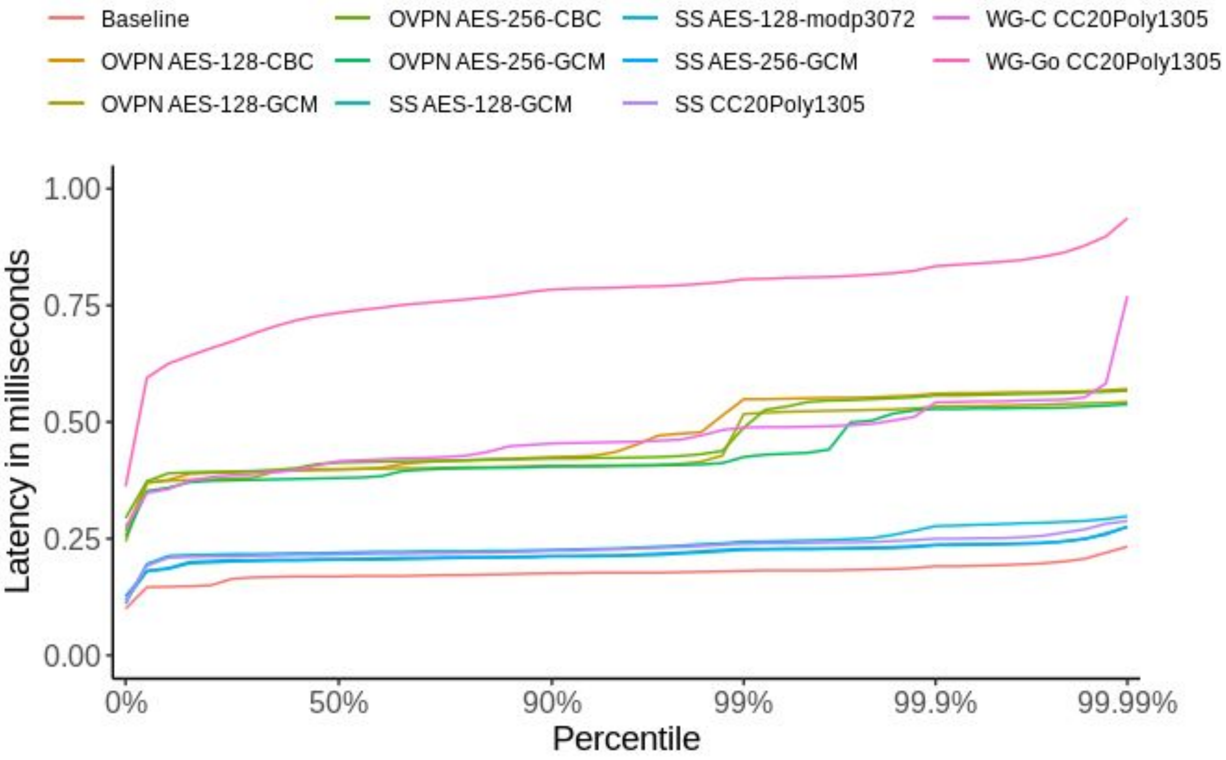
Conclusion

- In terms of TCP and UDP goodput, strongSwan is the best performing implementation, WireGuard-C follows closely behind. Overhead is the main limiting factor with maximum packet sizes.
- strongSwan has the lowest latency values, with WireGuard-C and OpenVPN performing equally. WireGuard-Go has the worst latency values by a large margin.
- Both WireGuard-C and WireGuard-Go are incredibly fast at initiating a connection. strongSwan is slightly slower, but not nearly as much as OpenVPN.
- strongSwan is the most efficient implementation in terms of CPU efficiency, while WireGuard-Go is the most inefficient.

Future work

- 10 Gbit/s environment
- iPerf alternatives such as Moongen
- Concurrent users
- Mobile environment
- ESP offloading
- Multi-threading

Questions?



Baseline
OVPN AES-128-CBC
OVPN AES-128-GCM
OVPN AES-256-CBC
OVPN AES-256-GCM
SS AES-128-GCM
SS AES-128-modp3072
SS AES-256-GCM
SS CC20Poly1305
WG-C CC20Poly1305
WG-Go CC20Poly1305
Latency in milliseconds
1.00
0.75
0.50
0.25
0.00
0%
50%
90%
99%
99.9%
99.99%
Percentile