JOURNAL
COMPUTER SCIENCES INSTITUTE

JCSI 27 (2023) 138–144
Received: 22 November 2022
Accepted: 18 December 2022

Comparative analysis of VPN protocols

Analiza porównawcza protokołów sieci VPN

Jerzy Antoniuk* , Malgorzata Plechawska-Wójcik

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract
The aim of the study was to check the performance of the set-up internet connection using the three VPN protocols: Wireguard, OpenVPN and L2TP / IPSec. The Docker applications were used for the tests, on which the VPN server configuration file was launched. The containers were running on the Amazon server. The tests were performed on a laptop and a virtual machine with Windows 10 Pro. The virtual machine has been run in the Microsoft Azure cloud. The next step was to launch three docker containers and start performance tests using three tools: ping command, Speedtest-cli and Iperf3. The result of the research is the analysis of the measurement results and drawing conclusions.

Keywords: computer network; cloud computing; virtual private network

Streszczenie
Celem badania było sprawdzenie wydajności zestawionego połączenia internetowego za pomocą trzech protokołów VPN: Wireguard, OpenVPN oraz L2TP/IPSec. Do badań wykorzystano aplikacje Docker, na której uruchomiono plik konfiguracyjny serwerów VPN. Kontenery były uruchomione na serwerze Amazon. Testy były wykonywane na laptopie oraz maszynę wirtualnej z systemem Windows 10 Pro. Maszyna wirtualna została uruchomiona w chmurze obliczeniowej Microsoft Azure. Kolejnym krokiem było uruchomienie trzech kontenerów dockerowych i rozpoczęcie testów wydajnościowych za pomocą trzech narzędzi: polecenie ping, program Speedtest-cli oraz Iperf3. Wynikiem badań jest analiza wyników pomiarów oraz wyciagnięcie wniosków.

Słowa kluczowe: sieci komputerowe; chmura obliczeniowa; wirtualna sieć prywatna

*Corresponding author
Email address: antoniukjurek@gmail.com (J. Antoniuk)

1. Wstęp

Sieci VPN coraz częściej towarzyszą ludziom w codziennych czynnościach. Szczególnie jest to widoczne od 2020, gdzie duża cześć sektora gospodarki rozpoczęła prace zdalną. W główniej mierze praca zdalna dotyczy branży IT, gdzie tworzone jest oprogramowanie dla firm i instytucji. W takim przypadku sieć VPN jest wykorzystywana do pracy zdalnej w domu. Coraz częściej osoby prywatne korzystają z prywatnej sieci wirtualnej do przeglądania multimediów, niedostępnych z powodu blokady geograficznej.

Celem artykułu naukowego było przeprowadzenie analizy wydajnościowej trzech wybranych protokół sieci VPN.

Teza, która została postawiona przez autora to „Protokół WireGuard jest wydajny w zastosowaniach konsumenckich".

W artykule postawiono następujące szczegółowe hipotezy badawcze:

1. Protokół WireGuard jest wydajniejszy niż starsze protokoły VPN.
2. Protokół WireGuard powoduje mniejsze obciążenie sieci komputerowej.
3. Protokół WireGuard zapewnia stabilne połączenie pomiędzy hostami.

Zakres artykułu obejmuje porównanie wydajności łącza z wykorzystaniem trzech protokołów VPN OpenVPN, Wireguard, L2TP/IPSec. W kolejnych krokach wykonano testy wydajnościowe i analizę otrzymanych wyników.

2. Przegląd literatury

W literaturze naukowej można spotkać wiele badań na temat wydajności sieci VPN. Analizę taką wykonuje się, aby sprawdzić czy jakość usług oferowanych przez wybrany serwer VPN będzie zadowalająca w określonych sytuacjach np. wykonywania zdalnych spotkań dla pracowników firmy. Badania takie również wykonuje się w przypadku, gdy firma będzie musiała wybrać optymalne rozwiązanie, które musi być niezawodnie i wydajnie. Jednak w przytoczonych poniżej badaniach naukowych można zobaczyć, że badane są również inne aspekty wpływające na wydajność połączenia np. opóźnienie pakietów, utrata wysyłanych ramek danych itp.

Najczęściej, analiza wydajności sieci VPN jest wykonywana w celu sprawdzenie maksymalnej przepustowości sieci komputerowej podczas korzystania z wybranego protokołu VPN [1, 2]. Takie analizy są zwykle wykonywane, aby można było stwierdzić jaki protokół VPN będzie optymalny np. do łączenia sieci komputerowych dwóch oddziałów firmy oddalonych o 1000 km. Jednak niektóre badania obejmują szerszy zakres kryteriów niż tylko maksymalna osiągana prędkość transmisji danych [2, 3, 4].

W artykule [5], mierzono wydajność protokołów VPN w dwóch wersji protokołów IP, w którym przedstawiono metody pomiaru przepustowości łącza z wykorzystaniem protokołu OpenVPN oraz WireGu-

ard. W tym przypadku brano pod uwagę trzy kryteria eksperymentu. Pierwszym kryterium było łącze 100Mbps przy każdej wykonywanej próbie. Kolejnym kryterium był czas każdego testu, który wynosił 10 sekund. Ostatnim warunkiem, który miał być spełniony był stopień utraty mniejszy poniżej 1%. Do tej próby wykorzystano dwa komputery, jeden pracujący pod systemem Windows 10 Pro i drugi serwer posiadający z systemem Ubuntu 19.04 Serwer. Pomiary wykonywano za pomocą programu Iperf [1, 6, 7]. Po wykonaniu eksperymentów okazało się, że tym wypadku bardziej wydajnym protokołem był Wireguard, co związane było przede wszystkim z wielkością bufora na serwerze.

Inne podejście do badań w zakresie wydajności protokołów VPN zaprezentowano w pracy [2]. Autorzy analizowali różnice wynikające z wysyłania przez sieć komputerową zróżnicowanych pakietów danych z zakresu od 1000 do 9000 bajtów. Analizę przeprowadzono z wykorzystaniem programu do pomiaru przepustowości Iperf3, podobnie jak w pracy [1]. Tak jak w pierwszej próbie wykorzystano dwa komputery połączone ze sobą. W tym eksperymencie mierzono również opóźnienie przy transmisji danych pakietów za pomocą popularnego narzędzia Iperf3. Po wykonaniu symulacji okazało się, że wielkość pakietów ma wpływ na opóźnienie transmisji sygnałów.

Ostatnim przedstawionym eksperymentem [3] była analiza wydajności sieci VPN w chmurze obliczeniowej. Jest to ważne, ponieważ coraz więcej usług dużych firm korzysta z chmury publicznej. W tym przypadku posłużono się programem komputerowym do wykonywania symulacji OPNET Simulator [4]. Badano trzy warianty połączenia dwunastu komputerów z dwoma serwerami VPN. Pierwszy wariant zakładał połączenie pomiędzy klientami a serwerem pocztowym bez zapory sieciowej i połączenia przez VPN. W kolejnych wariantach brano pod uwagę połączenie przez firewall lub skorzystanie z połączenia tunelowanego oraz włączonego filtrowania ruchu sieciowego. Po wykonaniu eksperymentów który brały wyżej wymienione kryteria oceny eksperymentów, okazało się, że włączenie firewall i VPN zmniejsza wydajność sieci i zwiększa opóźnienia przesyłanych pakietów [10].

Najczęstszym spotkanym rozwiązaniem analizy wydajności połączenia VPN jest połączenie dwóch komputerów ze sobą za pomocą sieci przewodowej. W tym wypadku jedna maszyna pełni rolę serwera usługi VPN a drugie urządzenie - klienta usługi. Najczęściej wtedy wykorzystuje się narzędzie Iperf, które pozwala na pomiar uzyskanych przepustowości zestawionego połączenia. Dobrym przykładem takiej analizy będzie porównanie wydajności protokołu OpenVPN [10, 13]. W tym przypadku połączono dwa komputery stacjonarny z przełącznikiem, Oba hosty miały dostęp do Internetu. W celu przedstawienia rzetelnych i powtarzalnych wyników ustawiono przepustowość łącza na 10Mbps. Komputery były połączone do tej samej podsieci. W tym eksperymencie skupiono się na sprawdzeniu wydajności pakietów przy protokole TCP oraz UDP [10]. Okazało się, że wydajność TCP jest trochę niższa, z powodu enkapsulacji pakietów. Również zauważono, że proces enkapsulacji i dekapsulacji ma wpływ na opóźnienie wysyłania i odbierania pakietów [7, 1].

Bardzo podobnym doświadczeniem [14], które zostało przeprowadzone w sieci dwóch uniwersytetów w Australii było porównanie wydajności łącza oraz opóźnienia sieci. W tym doświadczeniu skorzystano z protokołu IPSec. Aby było możliwe wykonanie tego przedsięwzięcia połączono ze sobą dwie bramki VPN połączone z Internetem na terenie tych uniwersytetów. Klienci VPN byli połączeni z lokalną siecią VPN za pomocą sieci bezprzewodowej. Zestawiono także połączenie pomiędzy typu Multiple Wireless VPN polegające na zestawieniu wielu połączeń jednocześnie [10, 14]. Następnie zrobiono pomiary prędkości i opóźnienia korzystając z oprogramowania Ipferf. Podczas tego eksperymentu sprawdzano prędkość transmisji podczas korzystania z jednego połączenia VPN jak i wielu. Również sprawdzono jakie opóźnienia generuje korzystanie z połączenia tunelowego. Przy okazji sprawdzono, ile procent wysłanych pakietów zostało utraconych podczas transmisji [3, 14]. Okazało się, że korzystanie z połączenia VPN miało duży wpływ na jakość wysyłanych paczek danych, co miało bezpośrednie przełożenie na prędkość i opóźnienia pakietów [9, 7].

3. Eksperyment badawczy

Do eksperymentu badawczego zostało użyte wbudowane do systemu Ubuntu oprogramowanie, ping. Do wykonywania testów przepustowości skorzystano z portali internetowych Speedtest-cli oraz dedykowanego oprogramowania Iperf3.

Podejmowane kroki do wykonania eksperymentu:

- Dziesięciokrotne wykonanie pomiarów prędkości połączenia internetowego za pomocą strony Speedtest-cli przed połączeniem VPN.
- Wykonanie 10 pomiarów za pomocą oprogramowania polecenia ping wbudowanych w system Windows 10.
- Wykonanie 10 pomiarów za pomocą testów jednokierunkowych i dwukierunkowych za pomocą oprogramowania Iperf z domyślnymi wartościami pakietów (segmentacja pakietów 100B, 150B i 200B) i różnymi wartościami pakietów.
- Połączenie się z serwerem VPN zlokalizowanym w chmurze Amazon za pomocą protokołu OpenVPN.
- Ponowne wykonanie pomiarów prędkości łącza za pomocą wcześniej wymienionych stron internetowych.
- Powtórzenie testów wykonywanych za pomocą oprogramowania Iperf z takimi samymi ustawieniami.
- Zmiana protokołu połączeniowego na L2TP/IPSec i ponowne wykonanie pomiarów poprzez przeglądarkę internetową i program Iperf z wcześniej wymienionymi założeniami.
- Zmiana protokołu połączeniowego na WireGuard i 10 krotne wykonanie pomiarów poprzez przeglądarkę firmy Google i program Iperf z wyżej wymienionymi założeniami.

4. Narzędzia badawcze

4.1. Narzędzia

Amazon Web Service

Amazon Web Services jest platformą usług przetwarzania w chmurze oferowaną przez firmę Amazon. Oferuje ona ponad 200 różnorodnych usług dla klientów na całym świecie. Firma udostępnia dla swoich klientów elementy takie jak: bazy danych, pamięć masowa, dedykowane oprogramowanie do uczenia maszynowego itp. Poziom usług oferowany przez firmę Amazon jest dosyć szeroki zaczynając od maszyn wirtualnych i kończąc na dedykowanym oprogramowaniu jako usłudze (ang. Software as a service*).* Dzięki temu można wykorzystać oferowane usługi przez firmę Amazon do różnorodnych zastosowań m.in. utrzymywania stron internetowych opartych na np. Wordpresie. W chmurze obliczeniowej AWS (ang. *Amazon Web Services*) można uruchomić każdy typ aplikacji m.in. aplikacje internetowa oparta na języku Java.

Z niektórych oferowanych usług Amazon Web Service można korzystać za darmo przez 12 miesięcy w ramach oferty Free Tier. Założeniem usługi Free Tier jest zdobywanie doświadczenia z korzystania z usług Amazon Web Services. Oferta zawiera następujące elementy:

- 750 godzin korzystania z maszyn wirtualnych z systemem Microsoft Windows oraz Linux,
- 750 godzin działania modułu równoważenia obciążenia,
- 5GB przestani dyskowej na dane,
- 20GB na bazę danych włącznie z kopiami zapasowymi,
- 2 miesiące działania maszyn wirtualnych odpowiedzialnych za uczenie maszynowe,
- 30 dni sieciowej ochrony antywirusowej.

Platforma Docker

Docker jest to platforma służąca do wdrażania, automatyzacji aplikacji uruchamianych w postaci kontenerów. W ten sposób można uruchamiać wiele aplikacji w postaci odizolowanych od siebie kontenerów. Głównym założeniem tej platformy jest eliminowanie problemów związanych z kompatybilnością aplikacji w środowisku produkcyjnym.

Platforma Docker działa w kontenerze Docker. W tym przypadku każdy obraz maszyny wirtualnej działa niezależnie od systemu operacyjnego gospodarza. Natomiast kontenery korzystają z jednego wspólnego jądra tego systemu operacyjnego działający na platformie hosta. Każdy kontener posiada własny system plików oraz zmienne środowiskowe, który są samowystarczalne do ich działania. Główną różnicą pomiędzy konteneryzacją a tradycyjną wirtualizacji jest wielkość zużywanej pamięci masowej. W tym wypadku kontener z reguły posiada rozmiar około rzędu kilkuset megabitów. W przypadku maszyn wirtualnych rozmiar obrazu jest rzędu kilku lub kilkunastu gigabitów. W ten sposób można uruchomić wiele kontenerów, które jednocześnie nie zużywają dodatkowych zasobów komputera np. pamięci operacyjnej. Zaletą kontenerów jest izolowanie każdej instancji pomiędzy sobą oraz systemem operacyjnym gościa. Pomaga to w zachowaniu wyższego bezpieczeństwa kontenerów. Drugą ważną cechą jaka wyróżnia konteneryzację w porównaniu do klasycznej wirtualizacji jest wydajność. Jest to związane z tym, że kontenery działają w ramach jednego jądra systemu operacyjnego na platformie Docker. Jedną z ważnych zalet architektury Dockera jest to, że kontenery po pobraniu obrazów są natychmiast uruchamiane.

Podsystem WSL

Podsystem Windows dla systemu Linux (WSL) (Podsystem Windows dla systemu Linux, 2017) jest to funkcja wprowadzona w rocznicowej aktualizacji systemu operacyjnego Windows 10 firmy Microsoft. WSL zapewnia pierwszą prawdziwie natywną obsługę aplikacji Linux w systemie operacyjnym Windows poprzez implementację ładowania i wykonywania aplikacji i bibliotek ELF. Pozwala ona na uruchamianie natywnych plików ELF, który zapewnia użytkownikom systemu Windows duży i zróżnicowany zestaw istniejących aplikacji linuksowych, takich jak serwery WWW, e-mail, FTP i SSH, a także pełny zestaw aplikacji dla użytkowników końcowych. Wraz z zapewnieniem prostej metody przenoszenia istniejących aplikacji z systemu Linux na Windows, firma Microsoft zobowiązała się również do długoterminowego wsparcia platformy WSL.

4.2. Sprzęt używany do badań

Specyfikacja sprzętu użytego do eksperymentu została przedstawiona w Tabelach 1—3, gdzie Tabela 1 opisuje serwer VPN, a Tabele 2 i 3 odpowiednio pierwszy i drugi komputer kliencki.

Tabela 1: Specyfikacja serwera VPN

Nazwa	Wartość
Procesor	Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz
Pamięć RAM	1GB DDR4
Dysk SSD	8GB
Karta sieciowa	Intel NUC 82371SB PIIX3 ISA [Natoma/Triton II]
System operacyjny	Ubuntu 20.04

Tabela 2: Specyfikacja techniczna komputera klienta mobilnego

Nazwa	Wartość
Procesor	Intel i5-8250u
Pamięć RAM	16GB DDR4
Dysk SSD	256GB
Karta sieciowa bezprzewodowa	Intel® Dual Band Wireless-AC 8265
System operacyjny	Windows 10 Pro 21H1

Tabela 3: Specyfikacja techniczna komputera klienta mobilnego

Nazwa	Wartość
Procesor	Intel Xeon Platinium 8168
Pamięć RAM	4GB DDR4
Dysk SSD	128GB
System operacyjny	Windows 10 Pro 20H2

Specyfikacja urządzenia dostępowego LTE Zyxel LTE4506-M606:

- modem kategorii 6;
- obsługa pasma LTE-FDD 1/3/7/8/20 (800/900/1800/2100/2600 MHz);
- obsługa pasma LTE-TDD 38/40 (2300/2600 MHz);
- obsługa DC-HSPA+/HSPA/UMTS/EDGE/GPRS/GSM;
- prędkość pobierania LTE do 300 Mb/s i prędkość wysyłania do 50 Mb/s;
- obsługa do 32 urządzeń bezprzewodowych;
- WLAN: 802.11a/b/g/n/ac, dwa pasma (2,4 GHz + 5 GHz);
- jedno złącze micro-USB do ładowania i jeden port Gigabit LAN.

Laptop został podłączony do sieci Wi-Fi rozgłaszającej sygnał na częstotliwości 5GHz na kanale 100. Do routera mobilnego połączone było tylko jedno urządzenie. Podczas wykonywana pomiarów skorzystano z aplikacji Speedtest-cli, Iperf3 oraz polecenia ping, który został wbudowany w jądra systemu Ubuntu. System Ubuntu 20.04 został uruchomiony pod systemem Windows 10 Pro za pomocą podsystemu WSL (ang. *Windows Subsystem for Linux*).

W przypadku pozostałych badań połączenie było realizowane do instancji serwera EC2 zlokalizowanego w chmurze Amazon. Fizycznie serwer był zlokalizowany w Europie a dokładniej w mieście Frankfurt. Klient stacjonarny, na którym wykonywano testy wydajnościowe serwerów VPN znajdował się w Centralnej części Stanów Zjednoczonych. Jednak w tym przypadku skorzystano z chmury firmy Microsoft. Było to podyktowane dostępnością systemu Windows 10 Pro z graficznym interfejsem użytkownika.

Pierwszy etapem gromadzenia wyników było zweryfikowanie działania łącza mobilnego LTE pod katem uzyskiwania zakładanych prędkości łącza. Jest to związane z problemem niestabilnej prędkości łącza mobilnego wynikającego z charakterystyki działania sieci komórkowej. W celu zniwelowania zmian prędkości, które mogły wpływać na wyniki, założoną prędkość łącza od klienta ustawiono na 50Mbit/s a klienta na 10Mbit/s. W przypadku łącza stacjonarnego wykonano dziesięciokrotnie test prędkości łącza programem Speedtest-cli z serwerem Orange Polska w Warszawie. W tym przypadku średnia prędkość Internetu wynosi 3100Mbit do klienta na 525 Mbit/s od klienta. Podczas przeprowadzania eksperymentu badawczego sprawdzano następujące parametry:

- upload – jest to przepustowość łącza, gdy użytkownik wysyła dane na serwer znajdujący się w sieci Internet;
- download - jest to prędkość łącza, gdy klient pobiera plik na swój komputer;
- opóźnienie – jest to czas jaki upływa pomiędzy wysłaniem pakietu przez serwer i dotarciem do odbiorcy;
- jitter – jest to zmiana opóźnienia przesyłania pakietu; parametr pozwala na sprawdzenie czy połączenie internetowe jest stabilne;
- packet lost – jest to procent pakietu danych, który został utracony podczas przesyłania ramki do hosta.

Po weryfikacji nastąpiła właściwa cześć przebiegu eksperymentu badawczego. Rozpoczęto pomiary prędkości łącza za pomocą narzędzia Speedtest-cli.

Drugim etapem przeprowadzania eksperymentu było przeprowadzenie testów za pomocą narzędzia Iperf3. W tej części pomiary były wykonywane dziesięciokrotne, gdzie sprawdzano wydajność łącza VPN pomiędzy dwoma hostami pomiędzy nadawcą i odbiorcą. Podczas testów założono wartości maksymalnej segmentacji pakietów na kolejno: 1000, 1500 i 2000 bitów. Za każdym razem powtarzano pomiar dla każdego badanego protokołu, aby wykluczyć wpływ zewnętrznych czynników.

Ostatnim krokiem było sprawdzenie połączenia pomiędzy serwerem VPN a klientem za pomocą programu ping.

W wierszu poleceń systemu Windows 10 mierzone opóźnienia pakietów transmitowane z komputera klienckiego do serwera, gdzie brano pod uwagę następujące parametry:

- minimalne opóźnienie,
- maksymalne opóźnienie,
- średnie opóźnienie,
- odchylenie standardowe opóźnienia,
- procentowa utrata pakietów,
- całkowity czas transmisji.

5. Wyniki badań

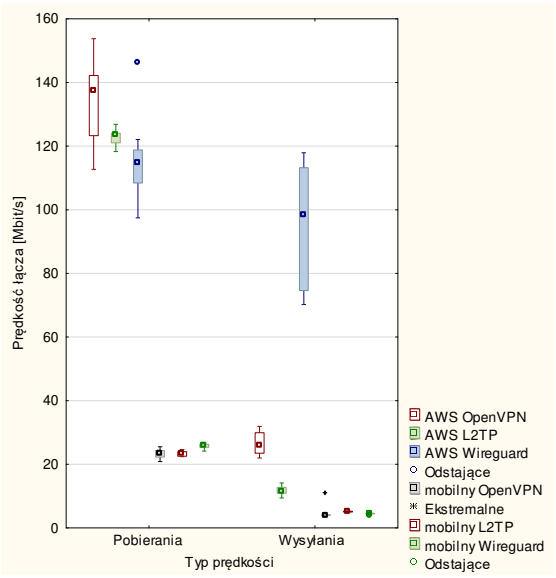
Badania zostały przeprowadzone za pomocą dwóch komputerów klienckich oraz maszyny wirtualnej znajdującej się w chmurze Amazon. Eksperyment polegał na przeprowadzeniu 10 testów za pomocą trzech programów: Speedtest-cli, Iperf3 oraz polecenia ping w wierszu poleceń systemu Ubuntu. System Ubuntu został uruchomiony na platformie WSL w systemie Windows 10 Pro. W tabeli 4 przedstawiono wyniki przepustowości połączenia pomiędzy serwerem a klientem, wykonane za pomocą programu Iperf3. Podczas przeprowadzania testów wydajności podzielono pomiary na dwie grupy określone przez: maksymalną segmentację pakietów o wartościach: 1000, 1500, 2000 bitów i drugą grupę określającą typ urządzenia klienckiego. Podczas wykonywania testów wydajnościowych zaobserwowano, że prędkość połączenia pomiędzy hostami w protokole Wireguard jest niższa o około 50% w stosunku do połączenia klienta stacjonarnego w pozostałych protokołach VPN. W pozostałych przypadkach

różnice pomiędzy prędkościami łącza internetowego wahają się w granicach 20% w zależności od protokołu VPN.

Tabela 4: Prędkości transmisji połączenia klienta z serwerem VPN za pomocą narzędzia Iperf3

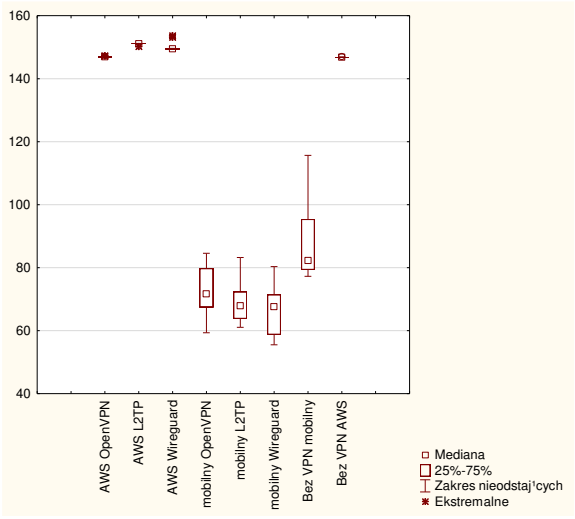
Nazwa	Nazwa protokołu	Urządzenie klienckie					
		stacjonarne			mobilne		
		Segmentacja pakietów					
		1000	1500	2000	1000	1500	2000
sender transfer (Mbits/sec)	Bez VPN	**10,84**	**10,84**	**11,05**	**12,56**	**12,53**	**12,56**
sender bitrate (Mbits/sec)		**9,07**	**9,07**	**9,27**	**10,55**	**10,53**	**10,52**
reciver transfer (Mbits/sec)		**10,78**	**10,78**	**10,96**	**12,53**	**12,52**	**12,52**
reciver bitrate (Mbits/sec)		**9,03**	**9,03**	**9,20**	**10,54**	**10,53**	**10,53**
sender transfer (Mbits/sec)	OpenVPN	**10,91**	**11,20**	**11,28**	**12,55**	**12,50**	**12,70**
sender bitrate (Mbits/sec)		**9,12**	**9,41**	**9,46**	**10,52**	**10,49**	**10,65**
reciver transfer (Mbits/sec)		**10,82**	**11,12**	**11,21**	**12,53**	**12,48**	**12,69**
reciver bitrate (Mbits/sec)		**9,05**	**9,34**	**9,41**	**10,52**	**10,48**	**10,65**
sender transfer (Mbits/sec)	L2TP/IPSec	**10,89**	**11,18**	**11,22**	**12,48**	**12,62**	**12,68**
sender bitrate (Mbits/sec)		**9,29**	**9,40**	**9,43**	**10,49**	**10,59**	**10,63**
reciver transfer (Mbits/sec)		**10,64**	**11,11**	**11,14**	**12,48**	**12,60**	**12,67**
reciver bitrate (Mbits/sec)		**9,05**	**9,34**	**9,36**	**10,49**	**10,60**	**10,62**
sender transfer (Mbits/sec)	Wireguard	**11,23**	**11,48**	**11,40**	**5,12**	**5,16**	**5,27**
sender bitrate (Mbits/sec)		**9,41**	**9,62**	**9,54**	**4,30**	**4,33**	**4,42**
reciver transfer (Mbits/sec)		**11,19**	**11,44**	**11,35**	**5,11**	**5,12**	**5,22**
* reciver bitrate (Mbits/sec)		**9,38**	**9,59**	**9,51**	**4,29**	**4,29**	**4,37**

Na rysunku 1 przedstawiono prędkość połączenia internetowego połączonego za pomocą sieci VPN. Wykres podzieloną na dwa grupy prędkości połączenia internetowego: prędkość pobierania i wysyłania. Na wykresie wyróżnia się prędkość wysyłania klienta stacjonarnego w protokole Wireguard, która waha się w granicach 110Mbit/s. Jednak warto zauważyć, że w pozostałych przypadkach prędkość wysyłania jest średnio 4-krotnie mniejsza niż w pozostałych pomiarach. W przypadku prędkości pobierania, różnice pomiędzy protokołami VPN pomiędzy jednym typem prędkości są niewielkie.



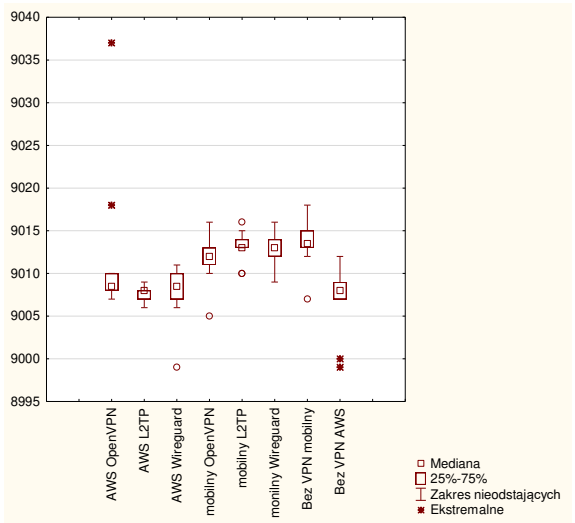
Rysunek 1: Wykres prędkości łącza za pomocą protokołów VPN.

Na rysunku (Rysunku 2) zaprezentowano wykres wartości opóźnienia sygnału zbadane za pomocą polecenia ping w systemie WSL. Jedyną zasadniczą różnicą zaobserwowaną na wykresie to wyższe opóźnienia w pomiarach wykonywanych za pomocą maszyny wirtualnej w chmurze Azure niż w przypadku pomiarów wykonywanych na komputerze mobilnym. Na laptopie opóźnienia transmisji osiągały wartości w zakresie od 1 do 20 ms a na stacjonarnym około 5 ms. W pozostałych przypadkach przedział wartości opóźnienia waha się w granicach 5ms.



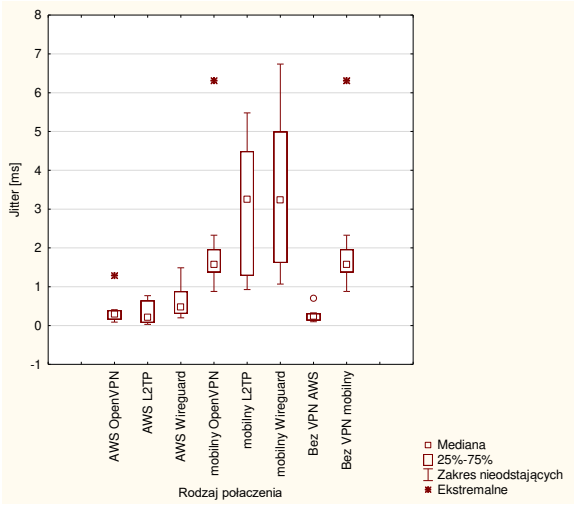
Rysunek 2: Wykres opóźnienia pakietów połączenia pomiędzy klientem i serwerem VPN za pomocą programu ping.

Na rysunku 3 zaprezentowano całkowity czas transmisji pakietu w komunikacji pomiędzy hostami. W tym przypadku pomiary wykonano za pomocą polecenia ping w systemie WSL. Generalnie całkowity czas transmisji jest niższy w przypadku klienta stacjonarnego. Jednak różnice pomiędzy całkowitym czasem transmisji pomiędzy komputerem stacjonarnym a laptopem mieszczą się w granicach 10ms.



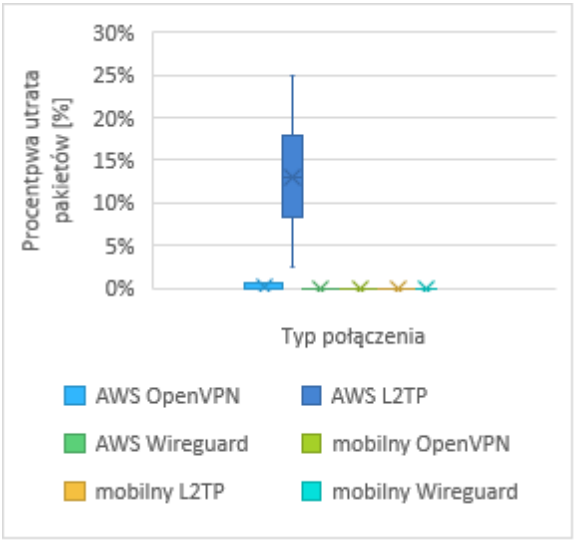
Rysunek 3: Wykres całkowitego czasu transmisji pomiędzy serwerem VPN a klientem w programie ping.

Na rysunku 4 przedstawiono wykres pudełkowy wariancji opóźnienia transmisji pakietów (ang. *Jitter*) w programie Iperf3. W tym przypadku można zauważyć duży zakres zmienności opóźnienia transmisji pakietów w przypadku protokołu OpenVPN oraz Wireguard. W tym przypadku przedział zmienności jittera wynosi 3ms. W przypadku pozostałych pomiarów zmienność opóźnienia transmisji pakietów mieści się w przedziale od 0 do 1,5 milisekund.



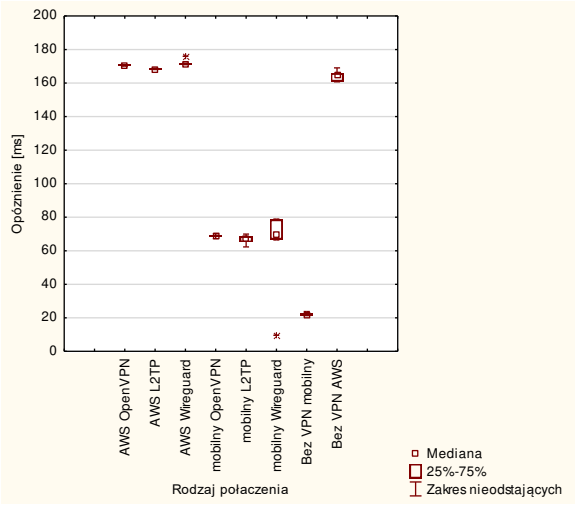
Rysunek 4: Wykres zmiany opóźnień pakietów połączenia (ang. Jitter) pomiędzy klientem i serwerem VPN w programie Iperf3.

Na rysunku 5 zaprezentowano wykres procentowej utraty pakietów pomiarów uzyskany z pomiarów wykonanych za pomocą narzędzia Speedtest-cli. W przypadku protokołu L2TP w komputerze stacjonarnym odnotowano stratę pakietów na poziomie 12,93%. Drugim protokołem, w którym zauważono problem z transmisją danych był OpenVPN. Odnotowano utratę pakietów dla komputera stacjonarnego na poziomie 0,29%. Dla pozostałych protokołów VPN nie wychwycono utraty pakietów.



Rysunek 5: Wykres procentowego utraty pakietów pomiędzy klientem i serwerem VPN w programie Speedtest-cli.

Na ostatnim rysunku 6 pokazano wyniki opóźnień transmisji pakietów danych zmierzonych za pomocą programu Speedtest-cli. W tym przypadku można zauważyć, że opóźnienia bez VPN jest niższe niż w pozostałych scenariuszach eksperymentu badawczego. W przypadku pomiarów wykonanych za pomocą protokołu Wireguard średnia wartość opóźnienia dla komputera mobilnego jest wyższa niż w innych protokołach VPN. Zakres opóźnienia dla protokołu Wireguard wynosi w granicach 15-20ms w przypadku pomiarów na komputerze mobilnym. W pozostałych przypadkach zakres waha się w przedziale 0-5ms.



Rysunek 6: Wykres zmiany opóźnień pakietów połączenia pomiędzy klientem i serwerem VPN w programie Speedtest-cli.

6. Wnioski

Przeprowadzone badania pozwoliły sprawdzić działanie sieci VPN w odniesieniu do wybranych protokołów VPN. Pierwszym spostrzeżeniem, które nasunęło się podczas testów wydajności było to, że prędkość wysyłania dla protokołu Wireguard, która oscylowała w zakresie 100Mbit-140Mbit/s. Warto zauważyć, że w pozostałych przypadkach prędkość połączenia do serwera oscylowała w granicach 20Mbit/s. Kolejnym elementem, który został zauważalny podczas korzystania z serwera VPN na domyślnych ustawieniach było to, że prędkość zestawionego połączenia VPN w protokole Wireguard. Przepustowość była mierzona za pomocą programu Iprerf3 i była ona mniejsza średnio o 50% w stosunku do klienta stacjonarnego. Podczas realizacji eksperymentu nie stwierdzono, czy było to spowodowane użytymi algorytmami do szyfrowania danych. Mogło to być związane konfiguracją stanowiska badawczego w protokole Wireguard. Sytuacja ta nie występowała w pozostałych protokołach VPN. Ważnym aspektem sieci komputerowej jest dostarczanie wszystkich pakietów danych. Jednak podczas mierzenia wydajności połączenia za pomocą programu Speedtest-cli odnotowano utratę pakietów w dwóch protokołach VPN: OpenVPN i L2TP.W pierwszym odnotowano średnio 13% a w drugim: 0,29%. Ostatni aspekt, który wpływa na wydajność łącza internetowego to zakres zmiany opóźnienia sygnału, które wahają się w granicach 10-

20ms dla klienta mobilnego. W pozostałych przypadkach przedział zmienności jitter wynosi około 5ms. Aby uzupełnić omówione analizy zostały także wykonane statystyczne testy nieparametryczne badające różnice pomiędzy poszczególnymi rodzajami połączeń. W badaniach użyto testu Kruskala-Wallisa z poziomem istotności ustawionym na wartość 0,05. W przypadku danych opóźnienia sygnału nie zauważono znaczących różnic pomiędzy wartościami opóźnienia dla poszczególnych grup. Zauważono istotne różnice pomiędzy następującymi grupami: prędkość pobierania połączenia, prędkość wysyłania, opóźnienie sygnału. Podczas wykonywania testów nieparametrycznych, nie stwierdzono istotnych różnic pomiędzy danymi zmienności opóźnienia sygnału i całkowitym czasem transmisji. Po przeprowadzonych eksperymentu badawczego można stwierdzić, że protokół Wireguard jest wydajny tylko w wybranych zastosowaniach. Jednak w celu weryfikacji wyników, należy ponownie wykonać eksperyment badawczy który pozwoli stwierdzić zależność pomiędzy odległością pomiędzy punktami końcowymi połączenia a prędkością łącza internetowego. Również należy wziąć pod uwagę konfiguracje serwerów VPN, które mogły wpływać na wydajność zestawionego połączenia sieciowego.

Literatura

[1] B. Hoekstra, D. Musulin, J. J. Keijser, Comparing TCP performance of tunneled and non-tunneled traffic using OpenVPN, Universiteit Van Amsterdam, System & Network Engineering, Amsterdam (2011) 2010-2011.

[2] C. A. Shue, M. Gupta, S. A. Myers, Ipsec: Performance analysis and enhancements, IEEE International Conference on Communications (2007) 1527-1532.

[3] S. Y. Ammen., S. W. Nourildean, Firewall and VPN investigation on cloud computing performance, International Journal of Computer Science and Engineering Survey 5(2) (2014) 15.

[4] M. Aamir, M. Zaidi, H. Mansoor, Performance Analysis of DiffServ based Quality of Service in a Multimedia Wired Network and VPN effect using OPNET, arXiv preprint rXiv:1206.5469 (2012).

[5] M. Sabbagh, A. Anbarje, Evaluation of WireGuard and OpenVPN VPN solutions, 2020.

[6] P. Dymora, M. Mazurek, T. Pilecki, Performance Analysis of VPN Remote Access Tunnels, Annales Universitatis Mariae Curie-Sklodowska sectio AI– Informatica 14 (3) (2014) 53-64.

[7] S. Ullah, J. Choi, H. Oh, IPsec for high speed network links: Performance analysis and enhancements, Future Generation Computer Systems 107 (2020) 112-125.

[8] M. Pudelko, P. Emmerich, S. Gallenmüller, G. Carle, Performance Analysis of VPN Gateways, IFIP Networking Conference (Networking) (2020) 325-333.

[9] D. Taneja, S. S. Tyagi, Factors Impacting the Performance of Data Transferred Via VPN, International Journal of Innovative Technology and Exploring Engineering 8 (2019) 2962-2966.

[10] I. Coonjah, P. Clarel Catherine, K. M. S. Soyjaudah, Experimental performance comparison between TCP vs UDP tunnel using OpenVPN, 2015, International Conference on Computing Communication and Security IEEE (2015) 1-5.

[11] M. Sakib, J. Singh. Simulation Based Performance Analysis of IPSec VPN over IPv6 Networks, International Journal of Electronics and Information Engineering 12(2) (2020) 92-104.

[12] S. Mackey, I, Mihov, Alex Nosenko, F. Vega, Y. Cheng, A Performance Comparison of WireGuard and OpenVPN, Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy (2020) 162-164.

[13] E. Danilova, Comparing the performance of different VPN technologies with TLS/SSL, 2015.

[14] K. S. Munasinghe, VPN over a wireless infrastructure: evaluation and performance analysis. Diss, University of Western Sydney, 2005.

[15] I. Kotuliak, P. Rybár, P. Trúchly, Performance comparison of IPsec and TLS based VPN technologies, 9th International Conference on Emerging eLearning Technologies and Applications IEEE (2011) 217-221.

[16] J. Ronan, S. Davy, J. Rossebo, An Analysis of IPsec Deployment Performance in High and Low Power Devices, (2004).

[17] R. Munadi, D. D. Sanjoyo, D. Perdana, F. Adjie, Performance analysis of tunnel broker through open virtual private network, Telkomnika 17(3) (2019) 1185-1192.

[18] S. Padhiar, Sneha, P. Verma, A Survey on Performance Evaluation of VPN, International Journal of Engineering Development and Research 3(4) (2015) 516-519.

[19] R. Malik, R. Syal, Performance analysis of ip security vpn, International Journal of Computer Applications 8.4 (2010) 0975.

[20] N. Lewis, Memory forensics and the windows subsystem for linux, Digital Investigation 26 (2018) S3-S11.