

**Technologie sieciowe**  
**Sprawozdanie Lista nr 1 – Wstęp**

Autor: Mateusz Laskowski 236618

## 1. Cel

Celami zadania pierwszej listy jest zapoznanie się z podstawową strukturą Internetu w ówczesnych czasach. Z pomocą paru narzędzi programistycznych oraz programów, takich jak Ping, Tracerout oraz WireShark, mamy za zadanie:

- Sprawdzić ilości węzłów na trasie do (i od) konkretnych serwerów na świecie
- Zbadać wpływ wielkości pakietu na szybkość i trasę przesłania danych
- Przyjrzyć się skutkom fragmentacji pakietów
- Zbadać największy niefragmentowany pakiet
- Przeanalizować różnice pomiędzy krótkimi, a długimi trasami
- Określić „średnicę” Internetu (najdłuższa ścieżka)
- Wyszukać oraz zbadać trasy przechodzące przez sieci wirtualne  
(zdalne platformy „cloud computing”)

## 2. Realizacja

### 2.1 Opis programów

**Ping** - program używany w sieciach komputerowych służący do diagnozowania połączeń sieciowych. Umożliwia sprawdzenia istnienia połączenia pomiędzy hostami testującymi i testowanym oraz zmierzenie liczby zgubionych pakietów, czy opóźnień w transmisji danych. Ping korzysta z protokołu ICMP (Internet Control Message Protocol – Internetowy Protokół Komunikatów Kontrolnych), który wysyła pakiety ECHO\_REQUEST oraz odbiera pakiety ECHO\_REPLY . Ciekawostka. Blokowanie wysyłania pakietów ECHO\_REPLY jest jedną z powszechnych metod ochrony przed atakami w sieci.

**TraceRoute** - program służący do badania trasy pakietów w sieci. Opiera się na protokołach komunikacyjnych UDP () oraz ICMP. Program ma na celu uzyskania informacji o routerach na trasie do hostu docelowego. Wykorzystuje przy tym pole w wysyłanym pakiecie TTL (Time To Live). Wartość ta jest zmniejszana przy przechodzeniu przez kolejne routery na trasie. Jeżeli wartość TTL osiągnie zero to pakiet jest odrzucany przez router, który wysyła informację zwrotną w postaci komunikatu ICMP, gdzie uzyskuje adres IP routera na trasie. Rozwijając działanie, TTL startuje z wartością 1 i po dojściu do routera zmniejsza wartość o 1 (aktualna wartość = 0) co powoduje powrót. Następny krok zwiększa TTL o jeden (początkowa 2) i przy napotkaniu routera jego wartość jest zmniejszana o 1.

Przy wartości zerowej znów polecenie zwraca IP ostatniego routera gdzie wartość TTL uzyskała wartość 0. Całe te działanie powtarza się, aż polecenie zwróci IP hostu docelowego.

**WireShark** - program komputerowy, którego zadaniem jest przechwytywanie, a także ich dekodowanie i ewentualnie analizowanie danych przepływających w sieci. Taki typ programów nazywany jest sniffer'em. Głównie jest używany do śledzenia pakietów. Jedną z zalet WireShark'a jest wykorzystanie graficznego interfejsu użytkownika.

## 2.2 Ilość węzłów

Sprawdzam „pingi” dla dwóch różnych stron internetowych. Jedna ma serwer w Polsce, zaś druga w Irlandii.

facebook.com (185.60.216.35) located in **Ireland**

```
$ ping -c 5 -l 1 -s 1472 -t 11 -M do facebook.com
PING facebook.com (185.60.216.35) 1472(1500) bytes of data.
1480 bytes from 185.60.216.35: icmp_seq=1 ttl=54 time=65.1 ms
1480 bytes from 185.60.216.35: icmp_seq=2 ttl=54 time=64.7 ms
1480 bytes from 185.60.216.35: icmp_seq=3 ttl=54 time=65.2 ms
1480 bytes from 185.60.216.35: icmp_seq=4 ttl=54 time=65.6 ms
1480 bytes from 185.60.216.35: icmp_seq=5 ttl=54 time=64.1 ms

--- facebook.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 64.100/64.975/65.637/0.513 ms
```

wp.pl (212.77.98.9) located in **Poland**

```
$ ping -c 5 -l 1 -s 1472 -t 7 -M do wp.pl
PING wp.pl (212.77.98.9) 1472(1500) bytes of data.
1480 bytes from (212.77.98.9): icmp_seq=1 ttl=59 time=45.2 ms
1480 bytes from (212.77.98.9): icmp_seq=2 ttl=59 time=44.4 ms
1480 bytes from (212.77.98.9): icmp_seq=3 ttl=59 time=45.5 ms
1480 bytes from (212.77.98.9): icmp_seq=4 ttl=59 time=44.0 ms
1480 bytes from (212.77.98.9): icmp_seq=5 ttl=59 time=44.0 ms

--- wp.pl ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 44.080/44.693/45.541/0.659 ms
```

Jak widać na powyższych przykładach, różnice w prędkości przesyłaniu danych dla takich samych wielkości pakietów są różne odpowiednio od odległości geograficznych danych serwerów. Sygnał do serwera facebook.com może pokonać trasę w 11 węzłach, gdzie jest minimalną liczbą węzłów, aby nie utracić danych w trasie. Zaś sygnał, który chce się dostać do serwera w Polsce musi pokonać przynajmniej 7 węzłów. Jak odczytujemy ilość węzłów od naszego hosta do serwera?

Za pomocą flagi -t 'x', gdzie x to liczba minimalnych węzłów na trasie. Ilość węzłów w powrotną stronę obliczamy ze wzoru, który przedstawię na przykładzie wp.pl:

$t_{tl}=59$ , szukamy najbliższą potęgę 2, gdzie tu będzie 64.  $64-59=5$

Gdzie 5 jest ilością węzłów w powrotnej drodze. Widać, że ilość węzłów do serwera i od serwera są różne!

## 2.3 Szybkość pakietów

Badam szybkość przesyłania danych. W pierwszym wpisaniu komendy wywołuje pakiet wielkości 1000 bajtów.

```
biblicalstudy.ru (90.156.201.76) located in Russia

$ ping -c 5 -l 1 -s 1000 -t 10 -M do biblicalstudy.ru
PING biblicalstudy.ru (90.156.201.76) 1000(1028) bytes of data.
1008 bytes from masterhost.ru (90.156.201.76): icmp_seq=1 ttl=57 time=78.3 ms
1008 bytes from masterhost.ru (90.156.201.76): icmp_seq=2 ttl=57 time=77.0 ms
1008 bytes from masterhost.ru (90.156.201.76): icmp_seq=3 ttl=57 time=78.4 ms
1008 bytes from masterhost.ru (90.156.201.76): icmp_seq=4 ttl=57 time=77.8 ms
1008 bytes from masterhost.ru (90.156.201.76): icmp_seq=5 ttl=57 time=78.1 ms

--- biblicalstudy.ru ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4074ms
rtt min/avg/max/mdev = 77.006/77.980/78.493/0.526 ms
```

W drugim wpisaniu komendy wywołuje pakiet wielkości 1472 bajtów, czyli o 472 bajty większe.

```
$ ping -c 5 -l 1 -s 1472 -t 10 -M do biblicalstudy.ru
PING biblicalstudy.ru (90.156.201.47) 1472(1500) bytes of data.
1480 bytes from masterhost.ru (90.156.201.47): icmp_seq=1 ttl=57 time=83.9 ms
1480 bytes from masterhost.ru (90.156.201.47): icmp_seq=2 ttl=57 time=83.3 ms
1480 bytes from masterhost.ru (90.156.201.47): icmp_seq=3 ttl=57 time=82.5 ms
1480 bytes from masterhost.ru (90.156.201.47): icmp_seq=4 ttl=57 time=83.4 ms
1480 bytes from masterhost.ru (90.156.201.47): icmp_seq=5 ttl=57 time=83.9 ms

--- biblicalstudy.ru ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 82.507/83.431/83.939/0.522 ms
```

Analizując oba wywołania komendy, wnioskuję, że wielkość pakietu ma znaczenie w szybkości przesyłania danych. Jak widać time w obu przypadkach się różni, a na dodatek w większym pakiecie czas jest dłuższy. Im większa wielkość pakietu tym większy czas propagacji.

## 2.4 Fragmentacja pakietów

```
$ ping -c 5 -s 1000 -M do eska.pl
PING eska.pl (212.180.238.58) 1000(1028) bytes of data.
1008 bytes from 212.180.238.58: icmp_seq=1 ttl=56 time=15.1 ms
1008 bytes from 212.180.238.58: icmp_seq=2 ttl=56 time=20.9 ms
1008 bytes from 212.180.238.58: icmp_seq=3 ttl=56 time=18.7 ms
1008 bytes from 212.180.238.58: icmp_seq=4 ttl=56 time=20.0 ms
1008 bytes from 212.180.238.58: icmp_seq=5 ttl=56 time=18.1 ms

--- eska.pl ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 15.131/18.605/20.979/1.998 ms
```

```
$ ping -c 5 -s 1000 -l 1 eska.pl
PING eska.pl (212.180.238.55) 1000(1028) bytes of data.
1008 bytes from 212.180.238.55: icmp_seq=1 ttl=56 time=35.9 ms
1008 bytes from 212.180.238.55: icmp_seq=2 ttl=56 time=17.9 ms
1008 bytes from 212.180.238.55: icmp_seq=3 ttl=56 time=18.3 ms
1008 bytes from 212.180.238.55: icmp_seq=4 ttl=56 time=18.3 ms
1008 bytes from 212.180.238.55: icmp_seq=5 ttl=56 time=19.3 ms

--- eska.pl ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4013ms
rtt min/avg/max/mdev = 17.907/21.975/35.922/6.989 ms
```

```
$ sudo ping -c 5 -s 1000 -l 3 eska.pl
PING eska.pl (212.180.238.58) 1000(1028) bytes of data.
1008 bytes from 212.180.238.58: icmp_seq=1 ttl=56 time=20.2 ms
1008 bytes from 212.180.238.58: icmp_seq=2 ttl=56 time=20.3 ms
1008 bytes from 212.180.238.58: icmp_seq=3 ttl=56 time=20.2 ms
1008 bytes from 212.180.238.58: icmp_seq=4 ttl=56 time=20.6 ms
1008 bytes from 212.180.238.58: icmp_seq=5 ttl=56 time=20.0 ms

--- eska.pl ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 20.083/20.314/20.628/0.219 ms, pipe 3
```

```
$ sudo ping -c 5 -s 1000 -l 5 eska.pl
PING eska.pl (212.180.238.55) 1000(1028) bytes of data.
1008 bytes from 212.180.238.55: icmp_seq=1 ttl=56 time=22.4 ms
1008 bytes from 212.180.238.55: icmp_seq=2 ttl=56 time=22.1 ms
1008 bytes from 212.180.238.55: icmp_seq=3 ttl=56 time=21.1 ms
1008 bytes from 212.180.238.55: icmp_seq=4 ttl=56 time=22.4 ms
1008 bytes from 212.180.238.55: icmp_seq=5 ttl=56 time=21.3 ms

--- eska.pl ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 17.112/18.826/20.400/1.334 ms, pipe 5
```

W pierwszej próbie wysłane zostały pakiety wielkości 1000 bajtów z zastrzeżoną fragmentacją, średni czas przesyłu w tej próbie wynosi 18.6 ms. W drugiej próbie wymuszona została fragmentacja pakietów na 1, średni czas propagacji wyniósł: 21.9 ms. Trzecia próba to fragmentacja na 3, gdzie średni czas to 20.4 Ostatnia próba to fragmentacja na 5, średni czas przesyłu to 21.9 ms.

Obserwacje: Czas przesyłu pojedynczego pakietu rośnie wraz ze zwiększeniem liczby pakietów po fragmentacji. Wniosek: Podział pakietu ma wpływ na czas jego przesyłu, na im więcej elementów jest dzielony tym czas większy, ponieważ dużo czasu zabiera samo fragmentowanie większych pakietów.

## 2.5 Największy niefragmentowany pakiet

Wysłałem największy możliwy pakiet na stronie interia.pl, gdzie na żadnej innej stronie nie mogłem wpisać. Maksymalną wielkość pakietu jaką uzyskałem to 23672 bajtów.

```
$ ping -l 1 -c 5 -s 23672 interia.pl
PING interia.pl (217.74.65.23) 23672(23700) bytes of data.
23680 bytes from star.interia.pl (217.74.65.23): icmp_seq=1 ttl=58 time=286 ms
23680 bytes from star.interia.pl (217.74.65.23): icmp_seq=2 ttl=58 time=286 ms
23680 bytes from star.interia.pl (217.74.65.23): icmp_seq=3 ttl=58 time=290 ms
23680 bytes from star.interia.pl (217.74.65.23): icmp_seq=4 ttl=58 time=286 ms
23680 bytes from star.interia.pl (217.74.65.23): icmp_seq=5 ttl=58 time=285 ms

--- interia.pl ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 285.336/286.893/290.265/1.831 ms
```

Można zauważyć jak wielki jest czas przesyłu pojedynczego pakietu. W Internecie znalazłem, że można uzyskać maksymalną wielkość pakietu 35512 bajtów. Wydaje mi się, że moje ograniczenia sprzętowe, nie pozwalają mi na większe pakiety.

## 2.6 Krótkie i długie trasy

Badając wcześniej wiele stron, doszedłem do wniosku, że długość trasy jest zależny od odległości geograficznej danego serwera od naszego hosta. Oczywiście im większa długość trasy to wzrasta czas propagacji, czyli przesyłanie pakietów na odległe serwery zajmuje znacznie więcej czasu. Trzeba pamiętać, że znak domeny nie zawsze oznacza również w tym samym kraju serwer strony. Dobrym przykładem jest

teora.maori.nz (35.189.48.215) located in <b>United States</b>
--

gdzie domena oznacza Nowo Zelandzką stronę a serwer jest w Stanach Zjednoczonych, co daje nam duży szybszy czas przesyłu, ponieważ Nowa Zelandia geograficznie jest dalej położona niż Stany Zjednoczone od naszego hosta (Polska, okolice Wrocławia).

## 2.7 „Średnica Internetu”

teora.maori.nz (35.189.48.215) located in <b>United States</b>
--

```
$ ping -c 5 -l 1 -s 1472 -t 27 -M do teora.maori.nz
PING teora.maori.nz (35.189.48.215) 1472(1500) bytes of data.
1480 bytes from (35.189.48.215): icmp_seq=1 ttl=55 time=350 ms
1480 bytes from (35.189.48.215): icmp_seq=2 ttl=55 time=349 ms
1480 bytes from (35.189.48.215): icmp_seq=3 ttl=55 time=349 ms
1480 bytes from (35.189.48.215): icmp_seq=4 ttl=55 time=348 ms
1480 bytes from (35.189.48.215): icmp_seq=5 ttl=55 time=349 ms

--- teora.maori.nz ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 348.799/349.389/350.268/0.740 ms
```

W tym przykładzie, osiągamy trasę z 27 węzłami, co wydaje się naprawdę duża w porównaniu z poprzednimi wywołaniami w konsoli. Wydawać się by mogło, że znak domeny .nz oświadczy nam, że serwer jest w Nowej Zelandii. Badając IP strony w programie Traceroute, wykryłem, że serwer jest w

Stanach Zjednoczonych, a nie w Nowej Zelandii. Podczas poszukiwania średnicy internetu najlepiej jest badać najbardziej oddalone geograficznie miejsca, tak więc Nowa Zelandia, USA czy Japonia. Najdalszym odnalezionym serwerem jest serwer japońskiej strony tmall.com, która jest odległa o 30 węzłów.

## **2.8 Cloud Computing – Sieci wirtualne**

Większe wartości uzyskano tylko na stronach chińskich, gdzie wartości te mogą sięgać nawet 45 węzłów. Większość stron z Chin napotkanych podczas testów jest tak odległa, może to wskazywać na to że w Chinach istnieje sieć wirtualna gdzie droga jaką pokonują pakiety może być celowo wydłużana. Kolejną rzeczą która wskazuje na istnienie takiej sieci w Chinach jest to, że tamtejsze strony najszybciej blokowały możliwość wysyłania pakietów („pingowanie”), zdarzyło się to już nawet przy 4-5 próbach.

## **3.0 Wnioski**

Sieć komputerowa jest nie do zmierzenia mimo iż składa się ze skończonej liczby serwerów i użytkowników, największym problemem jest to że cały czas ona się zmienia i określenie „średnicy” internetu co do węzła jest niemożliwe gdyż nigdy nie możemy wykluczyć istnienia serwera, który znajdował by się dalej. Wpływ na pomiary dokonywane w testach miało wiele czynników, chociażby rodzaj połączenia internetowego, czy też obciążenie sieci w danej chwili. Należy również pamiętać, że siecią można manipulować na różne sposoby. Na przykład przy używaniu Traceroute zdarza się, że część IP serwerów jest niewidoczna (niedostępna). Tak więc wszystkie pomiary należy brać z pewną rezerwą, gdyż nawet takie same próby jedna po drugiej są w stanie dać odmienne wyniki.