Technologie Sieciowe

Lista 1 - Ping, Traceroute, WireShark

Karol Janic

marzec 2023

1 Ping

1.1 Opis programu

Jest to program służący do diagnozowania połączeń sieciowych. Korzysta z protokołu ICMP(Internet Control Message Protocol). Pozwala na sprawdzenie, czy istnieje połączenie pomiędzy hostami testującym i testowanym. Umożliwia on zmierzenie liczby zgubionych pakietów oraz opóźnień w ich transmisji. Dzieje się to poprzez wysłanie do danego hosta pakietów żądania echa i oczekiwanie na odpowiedź.

1.2 Przykładowe użycie

```
ping wmi.uni.wroc.pl -c 5

PING wmi.uni.wroc.pl (156.17.4.28) 56(84) bytes of data.
64 bytes from www.wmi.uni.wroc.pl (156.17.4.28): icmp_seq=1 ttl=49 time=16.2 ms
64 bytes from www.wmi.uni.wroc.pl (156.17.4.28): icmp_seq=2 ttl=49 time=16.7 ms
64 bytes from www.wmi.uni.wroc.pl (156.17.4.28): icmp_seq=3 ttl=49 time=16.0 ms
64 bytes from www.wmi.uni.wroc.pl (156.17.4.28): icmp_seq=4 ttl=49 time=29.1 ms
64 bytes from www.wmi.uni.wroc.pl (156.17.4.28): icmp_seq=5 ttl=49 time=19.5 ms

--- wmi.uni.wroc.pl ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 16.025/19.502/29.142/4.978 ms
```

1.3 Badanie liczby węzłów na trasie

Do tego celu należy wykorzystać TTL (Time To Live). Wartość ta jest ustawiana przez nadawcę (flaga -t) a każdy węzeł zmniejsza ją o 1. Gdy TTL osiągnie wartość 0 pakiet jest usuwany i nie dociera do adresata. Należy zauważyć, że liczby węzłów trasy do serwera oraz trasy z serwera nie muszą być równe. Aby wyznaczyć liczbę węzłów do serwera należy znaleźć minimalną wartość TTL przy ktorej ping odpowiada pozytywnie. Aby wyznaczyć liczbę węzłów z serwera należy odjąć otrzymaną wartość TTL od najbliższej potęgi 2 większej niż ta wartość.

```
ping wmi.uni.wroc.pl -c 1 -t 15
PING wmi.uni.wroc.pl (156.17.4.28) 56(84) bytes of data.

--- wmi.uni.wroc.pl ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms



ping wmi.uni.wroc.pl -c 1 -t 16
PING wmi.uni.wroc.pl (156.17.4.28) 56(84) bytes of data.
64 bytes from www.wmi.uni.wroc.pl (156.17.4.28): icmp_seq=1 ttl=49 time=15.0 ms

--- wmi.uni.wroc.pl ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 14.976/14.976/14.976/0.000 ms
```

1.4 Porównanie liczby skoków w zależności od lokalizacji serwera

Tabela poniżej porównuje wyniki eksperymentów dla kilku serwerów, które znajdują się w różnych lokalizacjach. Pomiary wykonano 10 razy dla każdego serwera. Adresy IP pochodzą ze strony https://public-dns.info.

Adres serwera	Lokalizacja serwera	Liczba skoków do	Liczba skoków od	Średni czas[ms]
79.110.192.233	Wrocław(Polska)	10	10	39.3 ± 4.4
89.161.27.8	Białystok(Polska)	12	11	42.9 ± 6.2
130.149.8.20	Berlin(Niemcy)	14	14	47.2 ± 5.2
195.76.192.131	Barcelona(Hiszpania)	23	18	92.6 ± 4.8
41.59.200.123	Arusha(Tanzania)	20	19	315.6 ± 40.7
118.176.201.3	Seul(Korea Południowa)	32	24	$374.5 \pm 30.$
114.23.146.151	Wellington(Nowa Zelandia)	29	27	439.0 ± 40.6
190.64.140.243	Montevideo(Urugwaj)	19	14	364.1 ± 30.9
216.194.28.69	Nowy Jork(USA)	25	23	257.0 ± 39.8

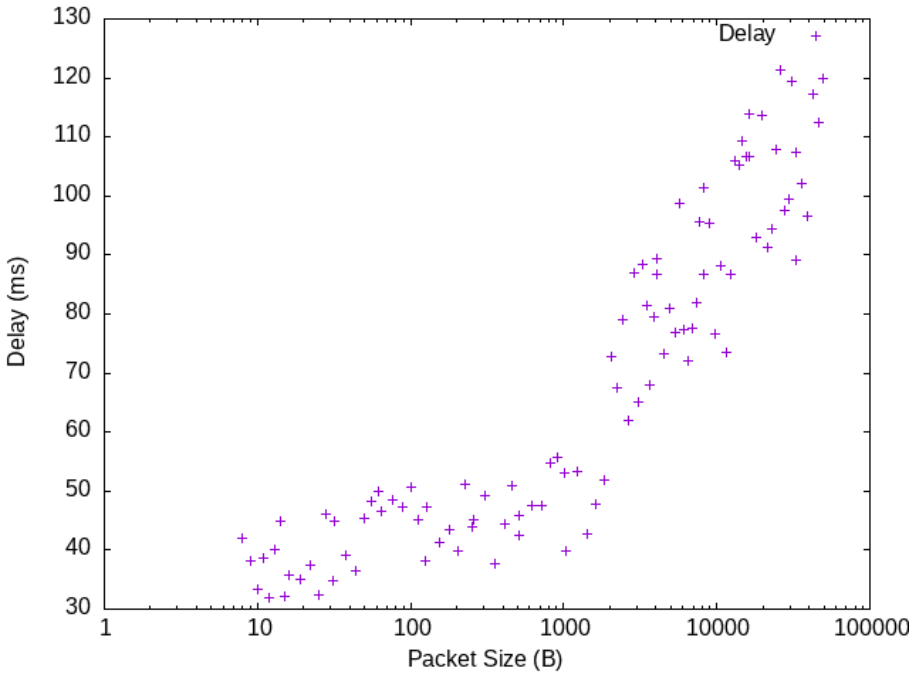
Wnioski:

- Najdalszy znaleziony serwer był w Seulu (32 węzły), więc jako "średnicę Internetu" możemy przyjąć 32.
- Największe średnie opóźnienie wygenerowało połączenie z Wellington, miastem w Nowej Zelandii. Wyniosło ono prawie 440 ms.
- W wielu eksperymentach liczba połączeń do i z serwera była różna. Jednym z powodów jest odmienna metoda przesyłania danych - gdzie liczby skoków różnią się nieznacznie(Arusha, Wellington, Nowy Jork). Kolejną przyczyną są sieci wirtualne. Można je rozpoznać po dużych rozbieżnościach w liczbie skoków i małych opóźnieniach(Barcelona, Seul, Montevideo).

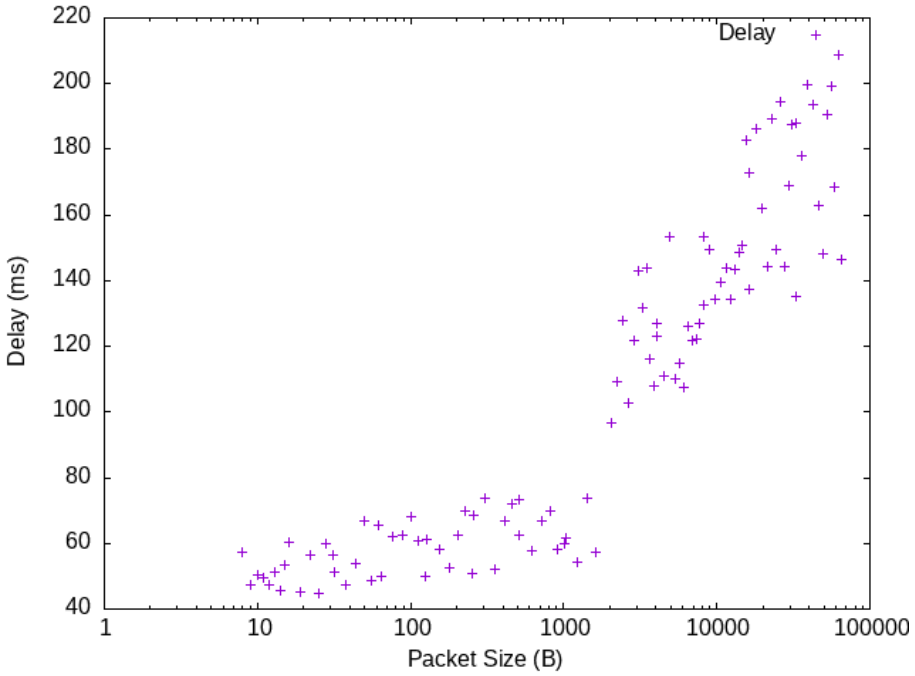
1.5 Badanie wpływu wielkości pakietu

Połączenie między poszczególnymi hostami ma ustaloną długość ramki, którą można przesłać. Jeśli jest ona zbyt długa, jest defragmentowana i wysyłana w kilku pakietach. Ma to negatywny wpływ na opóźnienie. Poniższy wykres przedstawia zależność opóźnienia od wielkości ładunku. Kod zamieszczony poniżej przedstawia skrypt wykonujący ten eksperyment.

```
#!/bin/bash

host="wmi.uni.wroc.pl"
packets_count=100
tmpfile=$(mktemp)

function ping_packet_size {
    ping -c $packets_count -s $1 -W 2 $host | tail -1 | awk '{print $4}'
    | cut -d '/' -f 2
}

for (( i=3; i<=15; i++ ))
do
    packet_size=$((2**$i))
    ping_time=$(ping_packet_size $packet_size)
    echo "$packet_size $ping_time" >> "$tmpfile"
done

gnuplot <<- EOF
set term png
set logscale x
set output "plot1.png"
set xlabel "Packet Size (B)"
set ylabel "Delay (ms)"
plot "$tmpfile" using 1:2 with points title "Delay"
EOF

rm "$tmpfile"
```



Rysunek 1: Średnie opóźnienie pingu do serwera we Wrocławiu w zależności od długości pakietu. Liczba skoków utrzymywała się na poziomie 14.



Rysunek 2: Średnie opóźnienie pingu do serwera we Wellington w zależności od długości pakietu. Liczba skoków utrzymywała się na poziomie 28.

Wnioski:

- Do pewnego rozmiaru danych opóźnienie rośnie powoli($1500B$).
- Po przekroczeniu wartości maksymalnej rozmiaru danych opóźnienie rośnie wraz ze wzrostem rozmiaru danych.

2 Traceroute

2.1 Opis programu

Jest to program służący do badania trasy pakietów w sieci przy użyciu TTL. Wysyła on zapytania do konkretnego hosta, za każdym razem zmieniając wartość TTL o jeden zaczynając od 0. Dzięki temu kolejne węzły wysyłają błędy przez co możemy je zidentyfikować.

2.2 Przykładowe użycie

```
traceroute to nasa.gov (52.0.14.116), 30 hops max, 60 byte packets
 1  _gateway (192.168.1.1)  2.643 ms  2.919 ms  2.906 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  ec2-52-0-14-116.compute-1.amazonaws.com (52.0.14.116)  19.017 ms  22.114 ms  25.152 ms


traceroute to cs.pwr.edu.pl (156.17.7.22), 30 hops max, 60 byte packets
 1  _gateway (192.168.1.1)  1.692 ms  3.484 ms  3.812 ms
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  informatyka.im.pwr.wroc.pl (156.17.7.22)  48.921 ms  50.797 ms  52.275 ms
```

Warto zauważyć, że niektóre linie są '* * *'. Oznacza to, że węzeł nie wysłał wiadomości, gdy TTL jest równe 0. Uniemożliwia to identyfikację poszczególnych węzłów.

3 WireShark

3.1 Opis programu

Jest to program służący do przechwytywania i analizy pakietów sieciowych(ale także USB). Może być używany do debugowania aplikacji sieciowych, ruchu sieciowego lub protokołów.

3.2 Przykładowe użycie

Poniższy ekran przedstawia analizę połączenia ze stroną internetową sklepu Robotronik. Możemy prześledzić jak działa komunikacja poprzez HTTP.