# Ping

Ping – polecenie używane w sieciach komputerowych TCP/IP (jak Internet) i służące do diagnozowania połączeń sieciowych. Pozwala na sprawdzenie, czy istnieje połączenie pomiędzy hostami testującym i testowanym. Umożliwia on zmierzenie liczby zgubionych pakietów oraz opóźnień w ich transmisji, zwanych lagami.

**1. Ile jest węzłów?**

```
push@push:~$ ping google.com
PING google.com(waw02s08-in-x0e.1e100.net (2a00:1450:401b:803::200e)) 56 data bytes
64 bytes from waw02s08-in-x0e.1e100.net (2a00:1450:401b:803::200e): icmp_seq=1 ttl=54 time=45.7 ms
64 bytes from waw02s08-in-x0e.1e100.net (2a00:1450:401b:803::200e): icmp_seq=2 ttl=54 time=64.8 ms
64 bytes from waw02s08-in-x0e.1e100.net (2a00:1450:401b:803::200e): icmp_seq=3 ttl=54 time=64.3 ms
64 bytes from waw02s08-in-x0e.1e100.net (2a00:1450:401b:803::200e): icmp_seq=4 ttl=54 time=68.4 ms
64 bytes from waw02s08-in-x0e.1e100.net (2a00:1450:401b:803::200e): icmp_seq=5 ttl=54 time=73.8 ms
64 bytes from waw02s08-in-x0e.1e100.net (2a00:1450:401b:803::200e): icmp_seq=6 ttl=54 time=76.8 ms
64 bytes from waw02s08-in-x0e.1e100.net (2a00:1450:401b:803::200e): icmp_seq=7 ttl=54 time=79.1 ms
64 bytes from waw02s08-in-x0e.1e100.net (2a00:1450:401b:803::200e): icmp_seq=8 ttl=54 time=85.4 ms
64 bytes from waw02s08-in-x0e.1e100.net (2a00:1450:401b:803::200e): icmp_seq=9 ttl=54 time=90.3 ms
64 bytes from waw02s08-in-x0e.1e100.net (2a00:1450:401b:803::200e): icmp_seq=10 ttl=54 time=90.1 ms
64 bytes from waw02s08-in-x0e.1e100.net (2a00:1450:401b:803::200e): icmp_seq=11 ttl=54 time=99.0 ms
64 bytes from waw02s08-in-x0e.1e100.net (2a00:1450:401b:803::200e): icmp_seq=12 ttl=54 time=102 ms
^C
--- google.com ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11016ms
rtt min/avg/max/mdev = 45.758/78.379/102.419/15.551 ms
```

Żeby dowidzieć się ile mamy węzłów musimy wiedzieć liczby możliwych węzlów na trasie:
Dla Windows 95: 32
Dla Windows 98 oraz Linux: 64
Dla Windows 7+: 128
Widzimy że TTL(Time To Live) jest równy 54, także widzimy że wysłane jest 64 bytes, z tego mamy 64-54=10, to znaczy że mamy 10 węzłów na trasie.

**2.Jaki wpływ ma wielkość pakietu na ilość węzłów?**

```
push@push:~$ ping -s 20 google.com
PING google.com(waw02s18-in-x0e.1e100.net (2a00:1450:401b:808::200e)) 20 data bytes
28 bytes from waw02s18-in-x0e.1e100.net (2a00:1450:401b:808::200e): icmp_seq=1 ttl=54 time=115 ms
```

```
push@push:~$ ping -s 100 google.com
PING google.com(waw02s18-in-x0e.1e100.net (2a00:1450:401b:808::200e)) 100 data bytes
76 bytes from waw02s18-in-x0e.1e100.net (2a00:1450:401b:808::200e): icmp_seq=1 ttl=54 (truncated)
```

```
push@push:~$ ping -s 20 wikipedia.org
PING wikipedia.org(text-lb.esams.wikimedia.org (2620:0:862:ed1a::1)) 20 data bytes
28 bytes from text-lb.esams.wikimedia.org (2620:0:862:ed1a::1): icmp_seq=1 ttl=55 time=102 ms
```

```
push@push:~$ ping -s 100 wikipedia.org
PING wikipedia.org(text-lb.esams.wikimedia.org (2620:0:862:ed1a::1)) 100 data bytes
108 bytes from text-lb.esams.wikimedia.org (2620:0:862:ed1a::1): icmp_seq=1 ttl=55 time=127 ms
```

Za pomocą polecenia -s (Linux) mogę zwiększyć albo zmniejszyć rozmiar pakietu.

Podczas różnych pomiarów widzimy że ilość węzłów nie zmieniła się, dlatego można wywnioskować że rozmiar pakietu nie wpływa na ilość węzłów na trasie.

**3.Jak wielkość pakietu wpływa na czas propagacji?**

`Propagation delay = distance/transmission speed = d/s`

Ze zdjęcia wyżej widzimy że czas propagacji rośnie.

### 4.Jaki największy niefragmentowany pakiet uda się przesłac?

W moim przypadku maksymalny rozmiar niefragmentowanego pakietu wynosi 1452

```
push@push:~$ ping -s 1452 google.com
PING google.com(waw02s18-in-x0e.1e100.net (2a00:1450:401b:808::200e)) 1452 data bytes
76 bytes from waw02s18-in-x0e.1e100.net (2a00:1450:401b:808::200e): icmp_seq=1 ttl=54 (truncated)
76 bytes from waw02s18-in-x0e.1e100.net (2a00:1450:401b:808::200e): icmp_seq=2 ttl=54 (truncated)
^C
--- google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 23.337/24.194/25.051/0.857 ms
push@push:~$ ping -s 1453 google.com
PING google.com(waw02s18-in-x0e.1e100.net (2a00:1450:401b:808::200e)) 1453 data bytes
^C
--- google.com ping statistics ---
81 packets transmitted, 0 received, 100% packet loss, time 81920ms
```

bajtów.

### 5. Średnica internetu

Średnica internetu to najdłuższa scieżka do serwera. W moim przypadku to jest strona australijska.

```
push@push:~$ ping immi.homeaffairs.gov.au
PING immi.homeaffairs.gov.au(www.homeaffairs.gov.au (2400:b300::143)) 56 data bytes
64 bytes from www.homeaffairs.gov.au (2400:b300::143): icmp_seq=1 ttl=41 time=535 ms
64 bytes from www.homeaffairs.gov.au (2400:b300::143): icmp_seq=3 ttl=41 time=560 ms
64 bytes from www.homeaffairs.gov.au (2400:b300::143): icmp_seq=4 ttl=41 time=393 ms
64 bytes from www.homeaffairs.gov.au (2400:b300::143): icmp_seq=5 ttl=41 time=607 ms
64 bytes from www.homeaffairs.gov.au (2400:b300::143): icmp_seq=6 ttl=41 time=396 ms
^C
--- immi.homeaffairs.gov.au ping statistics ---
6 packets transmitted, 5 received, 16% packet loss, time 5023ms
rtt min/avg/max/mdev = 393.549/498.758/607.646/87.823 ms
```

Widać że scieżka wynosi 23 węzły

### 6. Sieci Wirtualne

Sieci wirtualne modyfikują wartość wskaźnika TTL, przez co utrudnione jest śledzenie pakietów. To że nasz pakiet na swojej drodze przechodzi przez sieć wirtualną można rozpoznać po tym, że pingując kilka razy (z odstępami czasowymi) dostajemy różne wyniki TTL, lub odpowiedź uzyskujemy z różnych adresów IP.

## Traceroute

Traceroute sprawdza przez jakie komputery przepływają pakiety danych wysłane przez nasz komputer do wybranego serwera. Polecenie traceroute wypisze też czasy przechodzenia danych na poszczególnych odcinkach ich sieciowej drogi.

```
push@push:~$ traceroute google.com
traceroute to google.com (216.58.215.110), 30 hops max, 60 byte packets
 1  _gateway (192.168.0.1)  4.558 ms  4.503 ms  4.727 ms
 2  * * *
 3  pl-ktw01a-rc1-ae-18-0.aorta.net (84.116.253.129)  108.703 ms  111.081 ms  113.507 ms
 4  pl-waw26b-rc1-ae-40-0.aorta.net (84.116.133.29)  144.719 ms  146.564 ms  146.542 ms
 5  pl-waw26b-ri1-ae-24-0.aorta.net (84.116.138.73)  120.791 ms  123.490 ms  123.441 ms
 6  72.14.222.250 (72.14.222.250)  123.424 ms  103.300 ms  103.259 ms
 7  108.170.250.193 (108.170.250.193)  101.219 ms  113.564 ms  113.499 ms
 8  108.170.234.245 (108.170.234.245)  113.467 ms  127.592 ms  129.370 ms
 9  waw02s17-in-f14.1e100.net (216.58.215.110)  137.418 ms  137.419 ms  137.394 ms
```

Widzimy że traceroute pokonał 9 węzłów.

# Wireshark

Darmowe oprogramowanie open-source służące do analizowania pakietów. Działa w sposób pasywny, tzn. nie wysyła żadnych informacji, a tylko przechwytuje dane docierające do interfejsu sieciowego. Nie wpływa także w żaden sposób na działanie aplikacji przesyłających dane przez sieć. W aplikacji możemy używać różnych filtrów które ułatwiają nam odczytywanie przechwyconych danych. Dzięki temu programowi można łatwo odczytać informacje o zastosowanych protokołach oraz ich budowę.

No: Numer ramki

Time: Czas przechwyconej ramki (od uruchomienia przechwytywania) w sekundach

Source: Adres źródła ramki

Destination: Adres adresata ramki

Protocol: Rodzaj użytego protokołu

Length: Długość ramki

Info: Informacja