Yauheni Tsyrkunovich.

Sprawozdanie

Ogólna informacja

ping -help

```
jeiniok@jeiniok:~$ ping -help
Usage: ping [-aAbBdDfhLnOqrRUvV64] [-c count] [-i interval] [-I interface]
            [-m mark] [-M pmtudisc_option] [-l preload] [-p pattern] [-Q tos]
            [-s packetsize] [-S sndbuf] [-t ttl] [-T timestamp_option]
            [-w deadline] [-W timeout] [hop1 ...] destination
Usage: ping -6 [-aAbBdDfhLnOqrRUvV] [-c count] [-i interval] [-I interface]
             [-l preload] [-m mark] [-M pmtudisc_option]
             [-N nodeinfo_option] [-p pattern] [-Q tclass] [-s packetsize]
             [-S sndbuf] [-t ttl] [-T timestamp_option] [-w deadline]
             [-W timeout] destination
```

Na zrzucie widać opcje komendy “ping”.

Przykład użycia:

```
jeiniok@jeiniok:~$ sudo ping ki.pwr.edu.pl -c 5 -i 0.1 -t 20
PING ki.pwr.edu.pl (156.17.7.22) 56(84) bytes of data.
64 bytes from 156.17.7.22 (156.17.7.22): icmp_seq=1 ttl=56 time=5.19 ms
64 bytes from 156.17.7.22 (156.17.7.22): icmp_seq=2 ttl=56 time=5.74 ms
64 bytes from 156.17.7.22 (156.17.7.22): icmp_seq=3 ttl=56 time=29.7 ms
64 bytes from 156.17.7.22 (156.17.7.22): icmp_seq=4 ttl=56 time=31.8 ms
64 bytes from 156.17.7.22 (156.17.7.22): icmp_seq=5 ttl=56 time=3.58 ms

--- ki.pwr.edu.pl ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 404ms
rtt min/avg/max/mdev = 3.587/15.218/31.867/12.749 ms
```

- **-c** – liczba pakietów.
- **-i** – czas oczekiwania pomiędzy wysyłaniem następnych pakietów(dla czasu mniej, niż 0.2 sekund trzeba wpisać **sudo**).
- **-t** – ile przeskroków będzie żył pakiet.
- **-s** – liczba bajtów do wysyłania.
- **-M do** – zakaz fragmentacji.
- …

Długość trasy

Żeby sprawdzić, ile jest węzłów na trasie do wybranego serwera, trzeba za pomocą opcji **-t <liczba węzłów>** znaleźć wartość pograniczną, n. p.:

```
jeiniok@jeiniok:~$ sudo ping ki.pwr.edu.pl -c 5 -i 0.1 -t 9 -s 1920
PING ki.pwr.edu.pl (156.17.7.22) 1920(1948) bytes of data.
From 156.17.33.1 (156.17.33.1) icmp_seq=1 Time to live exceeded
From 156.17.33.1 (156.17.33.1) icmp_seq=2 Time to live exceeded
From 156.17.33.1 (156.17.33.1) icmp_seq=3 Time to live exceeded
From 156.17.33.1 (156.17.33.1) icmp_seq=4 Time to live exceeded
From 156.17.33.1 (156.17.33.1) icmp_seq=5 Time to live exceeded

--- ki.pwr.edu.pl ping statistics ---
5 packets transmitted, 0 received, +5 errors, 100% packet loss, time 403ms

jeiniok@jeiniok:~$ sudo ping ki.pwr.edu.pl -c 5 -i 0.1 -t 10 -s 1920
PING ki.pwr.edu.pl (156.17.7.22) 1920(1948) bytes of data.
1928 bytes from 156.17.7.22 (156.17.7.22): icmp_seq=1 ttl=56 time=9.02 ms
1928 bytes from 156.17.7.22 (156.17.7.22): icmp_seq=2 ttl=56 time=53.4 ms
1928 bytes from 156.17.7.22 (156.17.7.22): icmp_seq=3 ttl=56 time=7.10 ms
1928 bytes from 156.17.7.22 (156.17.7.22): icmp_seq=4 ttl=56 time=12.6 ms
1928 bytes from 156.17.7.22 (156.17.7.22): icmp_seq=5 ttl=56 time=19.1 ms

--- ki.pwr.edu.pl ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 411ms
rtt min/avg/max/mdev = 7.108/20.270/53.413/17.070 ms
jeiniok@jeiniok:~$
```

Na zrzucie ekranu widać, że nadawając długość życia 10 otrzymuję odpowiedź, natomiast przy **ttl** 9 mam komunikat o przekraczeniu czasu życia.

Żeby dowiedzieć się, ile węzłów jest na drodze od serwera, trzeba od wartości **ttl** serwera (w tym przypadku 64) odjąć **ttl** z komunikatu odpowiedzi.

ki.pwr.edu.pl:

64-56=8.

Czas propagacji

Wynik komendy **sudo ping ki.pwr.edu.pl -c 25 -i 0.1 -s 100** itp

rtt Serwer, liczba bajtów	min	avg	max	mdev
ki.pwr.edu.pl, 100 bajtów	2.929	10.797	76.823	15.508
ki.pwr.edu.pl, 10000 bajtów	10.075	23.478	58.605	12.916
diamond.jp, 100 bajtów	232.987	235.971	251.889	4.035
diamond.jp, 10000 bajtów	238.266	254.897	348.383	23.295

Widać, że jest różnica pomiędzy wysyłaniem 100 bajtów i 10000 bajtów (pofragmentowany pakiet).

Ona wynosi kilka milisekund I różnicy prawie nie widać na odległych serwerach, natomiast na serwerze ki.pwr.edu.pl różnica jest poważna.

Rozmiar pakietu

```
jeiniok@jeiniok:~$ sudo ping diamond.jp -c 6 -i 0.01 -s 1472 -M do
PING diamond.jp (210.148.177.240) 1472(1500) bytes of data.
1480 bytes from 210.148.177.240 (210.148.177.240): icmp_seq=1 ttl=235 time=235 ms
1480 bytes from 210.148.177.240 (210.148.177.240): icmp_seq=2 ttl=235 time=240 ms
1480 bytes from 210.148.177.240 (210.148.177.240): icmp_seq=3 ttl=235 time=240 ms
1480 bytes from 210.148.177.240 (210.148.177.240): icmp_seq=4 ttl=235 time=236 ms
1480 bytes from 210.148.177.240 (210.148.177.240): icmp_seq=5 ttl=235 time=246 ms
1480 bytes from 210.148.177.240 (210.148.177.240): icmp_seq=6 ttl=235 time=236 ms

--- diamond.jp ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 86ms
rtt min/avg/max/mdev = 235.906/239.258/246.243/3.633 ms, pipe 6
jeiniok@jeiniok:~$ sudo ping diamond.jp -c 6 -i 0.01 -s 1473 -M do
PING diamond.jp (210.148.177.240) 1473(1501) bytes of data.
ping: local error: Message too long, mtu=1500
ping: local error: Message too long, mtu=1500
ping: local error: Message too long, mtu=1500
ping: local error: Message too long, mtu=1500
ping: local error: Message too long, mtu=1500
ping: local error: Message too long, mtu=1500

--- diamond.jp ping statistics ---
6 packets transmitted, 0 received, +6 errors, 100% packet loss, time 80ms
```

Tu widać, że maksymalny rozmiar niefragmentowanego pakietu jest 1472 bajtów.

```
jeiniok@jeiniok:~$ sudo ping diamond.jp -c 4 -i 0.01 -s 25152
PING diamond.jp (210.148.177.240) 25152(25180) bytes of data.
25160 bytes from 210.148.177.240 (210.148.177.240): icmp_seq=1 ttl=235 time=269 ms
25160 bytes from 210.148.177.240 (210.148.177.240): icmp_seq=2 ttl=235 time=257 ms
25160 bytes from 210.148.177.240 (210.148.177.240): icmp_seq=3 ttl=235 time=262 ms
25160 bytes from 210.148.177.240 (210.148.177.240): icmp_seq=4 ttl=235 time=265 ms

--- diamond.jp ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 70ms
rtt min/avg/max/mdev = 257.185/263.829/269.691/4.566 ms, pipe 4
jeiniok@jeiniok:~$ sudo ping diamond.jp -c 4 -i 0.01 -s 25153
PING diamond.jp (210.148.177.240) 25153(25181) bytes of data.

--- diamond.jp ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 49ms
```

Maksymalny rozmiar pakietu pofragmentowanego jest 25152 bajty.

Średnica internetu

Największą ilość węzłów, którą udało się mi wykryć, jest 256-230=26.

```
jeiniok@jeiniok:~$ ping diamond.jp
PING diamond.jp (210.148.177.240) 56(84) bytes of data.
64 bytes from 210.148.177.240 (210.148.177.240): icmp_seq=1 ttl=230 time=255 ms
64 bytes from 210.148.177.240 (210.148.177.240): icmp_seq=2 ttl=230 time=259 ms
64 bytes from 210.148.177.240 (210.148.177.240): icmp_seq=3 ttl=230 time=235 ms
```

Ciekawostki

- Wszystkie badania prowadziłem w **Linuksie**, który jest zainstalowany w **VirtualBox**. Mam natomiast Windows jako główny system operacyjny. Zauważyłem, że kiedy wpisuję **ping** jednocześnie w **Windows** I **Linux**, to w **Linuksie** zawsze miałem **ttl** na jedynkę mniejszy. Ja to rozumiem tak, że **VirtualBox** można traktować jako jeszcze jeden węzeł.
- **ttl**, który otrzymuję od serwera zawsze jest różny. Można wysyłać dane I otrzymać **ttl** n.p. 52, a już za kilka minut 49.
- Na niektóre serwery, takie jak n.p. **google.com** nie mogę wysyłać więcej, niż ileś danych, dla **google.com** to jest 68 bajtów.

```
jeiniok@jeiniok:~$ sudo ping google.com -c 4 -i 0.0001 -s 68
PING google.com (216.58.215.78) 68(96) bytes of data.
76 bytes from 216.58.215.78 (216.58.215.78): icmp_seq=1 ttl=52 time=12.9 ms
76 bytes from 216.58.215.78 (216.58.215.78): icmp_seq=2 ttl=52 time=11.4 ms
76 bytes from 216.58.215.78 (216.58.215.78): icmp_seq=3 ttl=52 time=11.7 ms
76 bytes from 216.58.215.78 (216.58.215.78): icmp_seq=4 ttl=52 time=14.7 ms

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 37ms
rtt min/avg/max/mdev = 11.480/12.734/14.767/1.296 ms, pipe 2, ipg/ewma 12.370/12.899 ms
jeiniok@jeiniok:~$ sudo ping google.com -c 4 -i 0.0001 -s 69
PING google.com (216.58.215.78) 69(97) bytes of data.
76 bytes from 216.58.215.78 (216.58.215.78): icmp_seq=1 ttl=52 (truncated)
76 bytes from 216.58.215.78 (216.58.215.78): icmp_seq=2 ttl=52 (truncated)
76 bytes from 216.58.215.78 (216.58.215.78): icmp_seq=3 ttl=52 (truncated)
76 bytes from 216.58.215.78 (216.58.215.78): icmp_seq=4 ttl=52 (truncated)

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 44ms
rtt min/avg/max/mdev = 11.601/14.643/17.896/2.520 ms, pipe 2, ipg/ewma 14.834/12.918 ms
```

Traceroute

Za pomocą traceroute łatwiej jest śledzić za ścieżką od komputera do serwera, ponieważ widoczne są wszystkie IP komputerów, routerów, serwerów pomiędzy moim PC a serwerem docelowym. Ponadto, to ułatwia śledzenie sieci wirtualnych, bo będzie widać, że w krótkim czasie zmieniła się porządna liczba IP.

```
jeiniok@jeiniok:~$ sudo traceroute -I cuni.cz
[sudo] password for jeiniok:
traceroute to cuni.cz (195.113.89.35), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  0.379 ms  0.258 ms  0.094 ms
 2  192.168.0.1 (192.168.0.1)  1.642 ms  1.547 ms  1.570 ms
 3  156.17.230.254 (156.17.230.254)  2.221 ms  3.354 ms  4.036 ms
 4  234.ds.pwr.wroc.pl (156.17.229.234)  2.992 ms  2.877 ms *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * 212.191.237.121 (212.191.237.121)  10.130 ms  13.377 ms
11  80.249.209.106 (80.249.209.106)  34.113 ms  37.202 ms  38.544 ms
12  195.113.69.53 (195.113.69.53)  32.156 ms  37.966 ms  37.512 ms
13  195.113.69.178 (195.113.69.178)  47.928 ms  47.798 ms  47.519 ms
14  195.113.89.35 (195.113.89.35)  32.333 ms  31.895 ms  31.693 ms
15  195.113.89.35 (195.113.89.35)  34.484 ms  35.682 ms  36.837 ms
```

WireShark

Program przeznaczony do analizy pakietów, przechodzących przez PC. Program ma dużo ciekawostek.