

# Sprawozdanie

Technologie sieciowe – lista 1

Aleksandra Malinowska, WPPT INF, 4 semestr, marzec 2019

## 1. Używane programy

### 1.1. Ping

Ping to program, który wysyła pakiet danych pod podany adres sieciowy oraz otrzymuje informację zwrotną. W systemie Windows program ping można uruchomić m.in. z opcjami: **-t** (wysyłanie pakietów dopóki działanie programu nie zostanie przerwane), **-n count** (liczba pakietów do wysłania), **-i TTL** (czas życia pakietu).

### 1.2. Traceroute

Traceroute (tracert w systemie Windows) to program pokazujący dokładną ścieżkę, jaką przebywa pakiet danych wysłany pod podany adres sieciowy. Wynikiem jego działania jest lista routerów, do których dociera po drodze do celu wraz z czasem, jaki zajmuje każdy skok między routerami.

### 1.3. WireShark

WireShark to program służący do analizy przesyłanych protokołów sieciowych. Pozwala kontrolować ruch w sieci poprzez śledzenie wszystkich pakietów danych wysyłanych i przysyłanych do komputera.

## 2. Wywołania i analiza wyników

### 2.1. Ping

#### 2.1.1. Badanie długości trasy

##### 2.1.1.1. Wywołania dla serwerów w Europie

Program ping uruchomiony bez dodatkowych opcji dla serwera w Paryżu zwraca poniższe dane.

```
>ping 5.158.200.10

Pinging 5.158.200.10 with 32 bytes of data:
Reply from 5.158.200.10: bytes=32 time=65ms TTL=240
Reply from 5.158.200.10: bytes=32 time=54ms TTL=240
Reply from 5.158.200.10: bytes=32 time=86ms TTL=240
Reply from 5.158.200.10: bytes=32 time=59ms TTL=240
```

Maksymalna wielkość TTL dla pakietu to 255. Odpowiedzi z serwera w Paryżu to 240, zatem możemy wnioskować, że pakiet w drodze powrotnej pokonał trasę 15 węzłów.

```
>ping 5.158.200.10 -t -i 12

Pinging 5.158.200.10 with 32 bytes of data:
Reply from 5.158.200.123: TTL expired in transit.
Reply from 5.158.200.123: TTL expired in transit.
Reply from 5.158.200.123: TTL expired in transit.
Reply from 5.158.200.123: TTL expired in transit.

>ping 5.158.200.10 -t -i 13

Pinging 5.158.200.10 with 32 bytes of data:
Reply from 5.158.200.10: bytes=32 time=52ms TTL=240
Reply from 5.158.200.10: bytes=32 time=54ms TTL=240
Reply from 5.158.200.10: bytes=32 time=48ms TTL=240
Reply from 5.158.200.10: bytes=32 time=54ms TTL=240
```

Uruchamiając jednak wielokrotnie program z opcją -i możemy poznać długość trasy w kierunku serwera. Po kilku próbach udaje się ustalić, że długość tej trasy wynosi 13 węzłów, ponieważ dla liczby 12 paczka wygasła. Zatem trasa w obie strony nie jest jednakowa.

Podobnie jest dla serwerów w Getxo, Hiszpania (62.99.77.54), gdzie TTL odpowiedzi z serwera wynosi 49, skąd wnioskujemy, że przebyta trasa prawdopodobnie wynosiła 15 (TTL początkowo mógł wynosić 64). Natomiast trasa przebyta do serwera to 17 węzłów.

#### 2.1.1.2. Wywołania dla serwerów na świecie – szukanie najdłuższej trasy

Znalezienie „średnicy” internetu wiąże się ze znalezieniem najdłuższej trasy między komputerami. W poniższej tabelce znajdują się dane zgromadzone na podstawie uruchamiania programu ping dla różnych serwerów.

Adres IP serwera	Lokalizacja serwera	Odległość od serwera	Odległość do serwera
103.19.173.140	Mackay, Australia	18 (TTL=46)	20
1.34.236.54	Tajpej, Tajwan	17 (TTL=111)	19
190.54.110.23	Arauco, Chile	21 (TTL=234)	19
200.40.255.100	Montevideo, Urugwaj	19 (TTL=45)	21
41.134.5.66	Johannesburg, RPA	13 (TTL=51)	19
104.244.42.129	San Francisco, USA	7 (TTL=57)	7
58.83.160.156	Beijing, Chiny	19(TTL=45)	26

Z powyższych danych możemy wnioskować, że „średnica” internetu to około 26 węzłów.

#### 2.1.1.3. Wywołania dla serwerów w Polsce

Wartości otrzymywane w wyniku uruchamiania programu ping dla serwerów w Polsce nie przekraczają 12 węzłów i nie są w żaden sposób zależne od odległości geograficznej od serwera, np. droga z komputera we Wrocławiu do serwera w Gdańsku (83.14.251.234) to 11 węzłów, do serwera w Lublinie (83.0.202.74) to 10, a do serwera we Wrocławiu (93.181.135.25) to 12 węzłów.

#### 2.1.1.4. Różne długości tras

Zazwyczaj pakiety wysyłane do serwerów przebywają za każdym razem trasę tej samej długości. Zdarza się jednak, że kolejne odpowiedzi od serwera mają różne wartości TTL. Dzieje się tak, albo dlatego, że pakiet przebywa za każdym

razem inną trasę, albo dlatego, że routery po drodze ustawiają nowe wartości TTL, a nie tylko zmniejszają ją o jeden, jak powinny.

```
>ping 114.114.115.119

Pinging 114.114.115.119 with 32 bytes of data:
Reply from 114.114.115.119: bytes=32 time=166ms TTL=88
Reply from 114.114.115.119: bytes=32 time=167ms TTL=64
Reply from 114.114.115.119: bytes=32 time=195ms TTL=65
Reply from 114.114.115.119: bytes=32 time=165ms TTL=91

Ping statistics for 114.114.115.119:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 165ms, Maximum = 195ms, Average = 173ms
```

## 2.1.2. Badanie wpływu wielkości pakietu na jego trasę

### 2.1.2.1. Przesyłanie z użyciem fragmentacji

Wysyłając do danego serwera coraz większe pakiety można zauważyć proporcjonalną zależność czasu pokonania trasy przez pakiet do jego wielkości. Średni czas drogi rośnie wraz ze wzrostem wielkości pakietu. Nie jest to jednak zależność stała. Łatwo zauważyć, że wraz ze wzrostem wielkości pakietu zmniejsza się powtarzalność wyników przy kolejnych uruchomieniach programu. Dla małych pakietów średni czas drogi jest prawie stały. Dla większych waha się nawet o +/-50ms.

Po wielokrotnym wywołaniu programu dla różnych wartości widać również, że długość trasy nie zmienia się w zależności od wielkości pakietu (stałe TTL=240). Nie dochodzi również do żadnej utraty danych.

Warto zauważyć, że wielkość pakietu nie jest nieograniczona. Próba przesłania pakietu większego niż około 1450 bajtów kończy się niepowodzeniem.

```
>ping 5.158.200.10 -l 64

Pinging 5.158.200.10 with 64 bytes of data:
Reply from 5.158.200.10: bytes=64 time=68ms TTL=240
Reply from 5.158.200.10: bytes=64 time=48ms TTL=240
Reply from 5.158.200.10: bytes=64 time=55ms TTL=240
Reply from 5.158.200.10: bytes=64 time=49ms TTL=240

Ping statistics for 5.158.200.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 48ms, Maximum = 68ms, Average = 55ms

>ping 5.158.200.10 -l 256

Pinging 5.158.200.10 with 256 bytes of data:
Reply from 5.158.200.10: bytes=256 time=96ms TTL=240
Reply from 5.158.200.10: bytes=256 time=59ms TTL=240
Reply from 5.158.200.10: bytes=256 time=61ms TTL=240
Reply from 5.158.200.10: bytes=256 time=55ms TTL=240

Ping statistics for 5.158.200.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 55ms, Maximum = 96ms, Average = 67ms

>ping 5.158.200.10 -l 1000

Pinging 5.158.200.10 with 1000 bytes of data:
Reply from 5.158.200.10: bytes=1000 time=65ms TTL=240
Reply from 5.158.200.10: bytes=1000 time=66ms TTL=240
Reply from 5.158.200.10: bytes=1000 time=222ms TTL=240
Reply from 5.158.200.10: bytes=1000 time=64ms TTL=240

Ping statistics for 5.158.200.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 64ms, Maximum = 222ms, Average = 104ms

>ping 5.158.200.10 -l 1024

Pinging 5.158.200.10 with 1024 bytes of data:
Reply from 5.158.200.10: bytes=1024 time=56ms TTL=240
Reply from 5.158.200.10: bytes=1024 time=53ms TTL=240
Reply from 5.158.200.10: bytes=1024 time=58ms TTL=240
Reply from 5.158.200.10: bytes=1024 time=56ms TTL=240

Ping statistics for 5.158.200.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 53ms, Maximum = 58ms, Average = 55ms

>ping 5.158.200.10 -l 1200

Pinging 5.158.200.10 with 1200 bytes of data:
Reply from 5.158.200.10: bytes=1200 time=65ms TTL=240
Reply from 5.158.200.10: bytes=1200 time=79ms TTL=240
Reply from 5.158.200.10: bytes=1200 time=55ms TTL=240
Reply from 5.158.200.10: bytes=1200 time=86ms TTL=240

Ping statistics for 5.158.200.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 55ms, Maximum = 86ms, Average = 71ms

>ping 5.158.200.10 -l 1450

Pinging 5.158.200.10 with 1450 bytes of data:
Request timed out.
```

Powyższe próby zostały przeprowadzone na trasie o średniej długości (15 węzłów). Podobnie zachowują się pakiety na dłuższych oraz krótszych trasach. W miarę wzrostu wielkości pakietu, można zaobserwować wydłużenie czasu odpowiedzi. Zależy on również od samej długości trasy.

Wyraźną zmianę w czasie przesyłania pakietu można zauważyć przy wysyłaniu bardzo dużej ilości danych. Szczególnie widać to przy przesyłaniu danych do routera, z którym bezpośrednio jest połączony nasz komputer.

```
>ping 192.168.0.1 -n 5

Pinging 192.168.0.1 with 32 bytes of data:
Reply from 192.168.0.1: bytes=32 time=4ms TTL=64
Reply from 192.168.0.1: bytes=32 time=9ms TTL=64
Reply from 192.168.0.1: bytes=32 time=4ms TTL=64
Reply from 192.168.0.1: bytes=32 time=4ms TTL=64
Reply from 192.168.0.1: bytes=32 time=6ms TTL=64

Ping statistics for 192.168.0.1:
    Packets: Sent = 5, Received = 5, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 4ms, Maximum = 9ms, Average = 5ms

>ping 192.168.0.1 -n 5 -l 65500

Pinging 192.168.0.1 with 65500 bytes of data:
Reply from 192.168.0.1: bytes=65500 time=28ms TTL=64
Reply from 192.168.0.1: bytes=65500 time=28ms TTL=64
Reply from 192.168.0.1: bytes=65500 time=26ms TTL=64
Reply from 192.168.0.1: bytes=65500 time=40ms TTL=64
Reply from 192.168.0.1: bytes=65500 time=26ms TTL=64

Ping statistics for 192.168.0.1:
    Packets: Sent = 5, Received = 5, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 26ms, Maximum = 40ms, Average = 29ms
```

#### 2.1.2.2. Przesyłanie z blokadą fragmentacji

Wysyłając wielokrotnie pakiety o różnych rozmiarach do serwera w Ashburn, USA (52.206.51.15) widać, że użycie fragmentacji nie ma wpływu na średni czas przesyłania. Przy użyciu fragmentacji nie dochodzi również do żadnej utraty danych, a droga pozostaje tej samej długości.

```
>ping 52.206.51.15 -l 500 -n 2

Pinging 52.206.51.15 with 500 bytes of data:
Reply from 52.206.51.15: bytes=500 time=160ms TTL=237
Reply from 52.206.51.15: bytes=500 time=166ms TTL=237

Ping statistics for 52.206.51.15:
    Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 160ms, Maximum = 166ms, Average = 163ms

>ping 52.206.51.15 -l 500 -n 2 -f

Pinging 52.206.51.15 with 500 bytes of data:
Reply from 52.206.51.15: bytes=500 time=151ms TTL=237
Reply from 52.206.51.15: bytes=500 time=146ms TTL=237

Ping statistics for 52.206.51.15:
    Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 146ms, Maximum = 151ms, Average = 148ms

>ping 52.206.51.15 -l 1024 -n 2 -f

Pinging 52.206.51.15 with 1024 bytes of data:
Reply from 52.206.51.15: bytes=1024 time=149ms TTL=237
Reply from 52.206.51.15: bytes=1024 time=147ms TTL=237

Ping statistics for 52.206.51.15:
    Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 147ms, Maximum = 149ms, Average = 148ms

>ping 52.206.51.15 -l 1024 -n 2

Pinging 52.206.51.15 with 1024 bytes of data:
Reply from 52.206.51.15: bytes=1024 time=149ms TTL=237
Reply from 52.206.51.15: bytes=1024 time=148ms TTL=237

Ping statistics for 52.206.51.15:
    Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 148ms, Maximum = 149ms, Average = 148ms
```

Wielkość największego niepofragmentowanego pakietu przesłanego do powyższego serwera wynosi 1432 bajty.

```
>ping 52.206.51.15 -l 1433 -n 1 -f

Pinging 52.206.51.15 with 1433 bytes of data:
Packet needs to be fragmented but DF set.

>ping 52.206.51.15 -l 1432 -n 1 -f

Pinging 52.206.51.15 with 1432 bytes of data:
Reply from 52.206.51.15: bytes=1432 time=149ms TTL=237
```

## 2.2. Traceroute

### 2.2.1. Sposób działania

Program `tracert` sprawdza trasę do serwera poprzez wysyłanie pakietu z ustaloną wartością TTL. Zaczyna od liczby 1 a kończy w momencie, gdy dotrze do celu. Każdą liczbę testuje trzy razy, po czym przechodzi do następnej.

### 2.2.2. Przykładowe wywołanie

Program `tracert` uruchomiony dla serwera w Paryżu (5.158.200.10) zwraca wynik w postaci listy 13 routerów, co odpowiada długości trasy do tego serwera obliczonej w rozdziale 2.1.1.1.

```
>tracert 5.158.200.10

Tracing route to 5.158.200.10 over a maximum of 30 hops

  1    4 ms    4 ms    4 ms  192.168.0.1
  2    *      *      *      Request timed out.
  3   26 ms   34 ms   27 ms  pl-ktw01a-rc1-ae18-0.aorta.net [84.116.253.129]
  4   77 ms   31 ms   29 ms  pl-waw26b-rc1-ae40-0.aorta.net [84.116.133.29]
  5   31 ms   30 ms   25 ms  pl-waw26b-ri1-ae24-0.aorta.net [84.116.138.73]
  6   31 ms   24 ms   35 ms  ae4.cr0-waw3.ip4.gtt.net [213.254.220.105]
  7  101 ms   44 ms   57 ms  et-2-3-0.cr2-ams1.ip4.gtt.net [213.200.117.42]
  8   48 ms   54 ms   62 ms  ae20-ams-koo-score-2.interoute-gw.ip4.gtt.net [77.67.94.142]
  9   46 ms   59 ms   47 ms  ae0-0.ams-koo-score-1-re0.interoute.net [84.233.190.1]
 10   86 ms   52 ms   47 ms  ae2-0.par-gar-score-2-re0.interoute.net [84.233.190.62]
 11   63 ms   50 ms   66 ms  84.233.144.138
 12   50 ms   50 ms   49 ms  5.158.200.123
 13   58 ms   67 ms   51 ms  5.158.200.10

Trace complete.
```

Wynik zgadza się również dla serwera we Wrocławiu (93.181.135.25), do którego trasa wynosi 12 węzłów (rozdział 2.1.1.3.).

```
>tracert 93.181.135.25

Tracing route to 93-181-135-25.internetia.net.pl [93.181.135.25]
over a maximum of 30 hops:

  1    5 ms    8 ms    *    192.168.0.1
  2    *      *      *      Request timed out.
  3   24 ms   31 ms   31 ms  pl-ktw01a-rc1-ae18-0.aorta.net [84.116.253.129]
  4   33 ms   24 ms   26 ms  pl-waw26b-rc1-ae40-0.aorta.net [84.116.133.29]
  5   25 ms   26 ms   23 ms  pl-waw02a-ri1-ae54-0.aorta.net [84.116.134.126]
  6   26 ms   31 ms   33 ms  83.238.248.189
  7   38 ms   38 ms   39 ms  83.238.249.64
  8   45 ms   35 ms   34 ms  jawoh001rt09.inetia.pl [83.238.251.99]
  9   40 ms   50 ms   40 ms  83.238.248.195
 10    *      *      *      Request timed out.
 11   38 ms  105 ms   38 ms  93-181-130-90.internetia.net.pl [93.181.130.90]
 12   44 ms   36 ms   38 ms  93-181-135-25.internetia.net.pl [93.181.135.25]

Trace complete.
```

### 2.2.3. Analiza wyników

W pierwszej linii wyniku zwracanego przez program `tracert` otrzymujemy informację o tym, do jakiego serwera wysyłamy dane oraz jaka jest maksymalna liczba skoków.

Następnie otrzymujemy tabelę, w której pierwsza kolumna jest kolejną wartością TTL pakietu. Kolumny od 2 do 4 przedstawiają czas odpowiedzi serwera dla kolejnych wysyłanych pakietów (te próby są 3). Ostatnia kolumna to nazwa i adres IPv4 routera, do którego pakiet dotarł przy danej wartości TTL.

W przypadku gdy otrzymujemy „Request timed out” oznacza to, że router nie odpowiadał na żądania, dlatego po 3 próbach połączenia, został wybrany następny.

#### 2.2.4. Trasa geograficzna

Trasa wyznaczona przez tracert do serwera w San Francisco (104.244.42.129) to 7 węzłów.

```
>tracert 104.244.42.129

Tracing route to 104.244.42.129 over a maximum of 30 hops

  1    4 ms    4 ms    4 ms  192.168.0.1
  2    *      *      *      Request timed out.
  3   35 ms   45 ms   51 ms  pl-ktw01a-rc1-ae18-0.aorta.net [84.116.253.129]
  4   52 ms   42 ms   31 ms  de-fra04d-rc1-ae30-0.aorta.net [84.116.137.41]
  5   38 ms   33 ms   37 ms  de-fra04c-ri1-ae9-0.aorta.net [84.116.140.190]
  6   35 ms   49 ms   34 ms  de-fra01a-ri2-xe-0-2-0.aorta.net [213.46.179.110]
  7   34 ms   43 ms   33 ms  104.244.42.129

Trace complete.
```

Przy użyciu strony [www.iplocation.net](http://www.iplocation.net) można sprawdzić orientacyjną lokalizację geograficznego danego adresu sieciowego. Według danych z tej strony pakiet przebył trasę: Wrocław -> Katowice -> Frankfurt nad Menem -> San Francisco. Zgodnie z danymi Google Maps jest to odległość około 10 tysięcy kilometrów.

Z drugiej strony trasa wyznaczona do serwera we Wrocławiu (93.181.135.25) to 12 węzłów (wynik działania programu tracert dla tego serwera w rozdziale 2.2.2.). Geograficznie pakiet przebył trasę: Wrocław -> Katowice -> Warszawa -> Dublin -> Warszawa -> Łódź -> Warszawa -> Wrocław co sumarycznie daje trasę około 5500 km.

Na tych dwóch przykładach wyraźnie widać, że odległość geograficzna między dwoma komputerami nie ma związku z długością trasy przebywaną przez pakiet.



### 2.3. WireShark

### 2.3.1. Przykładowe użycie

Program Wireshark wypisuje w oknie dialogowym w czasie rzeczywistym listę wszystkich przesyłanych pakietów danych od i do komputera. Program umożliwia filtrowanie wyników, co pozwala na wyszukiwanie niechcianego ruchu w sieci. Dzięki temu programowi można uzyskać również szczegółowe informacje na temat każdej paczki danych.

No.	Time	Source	Destination	Protocol	Length	Info
34229	903.614863	2a02:a317:e13b:8880...	2001:730:3ed2:1000...	DNS	97	Standard query 0x65cf AAAA fonts.gstatic.com
34230	903.614969	2a02:a317:e13b:8880...	2001:730:3ed2:1000...	DNS	100	Standard query 0x0676 AAAA fonts.googleapis.com
34236	903.633994	2001:730:3ed2:1000...	2a02:a317:e13b:8880...	DNS	152	Standard query response 0x46c5 A fonts.googleapis.com CNAME googledadspl.l.google.com A 216.58.215.106
34237	903.642422	2001:730:3ed2:1000...	2a02:a317:e13b:8880...	DNS	161	Standard query response 0x65cf AAAA fonts.gstatic.com CNAME gstaticadsl.l.google.com AAAA 2a00:1450:401b:804:2003
34238	903.643536	2001:730:3ed2:1000...	2a02:a317:e13b:8880...	DNS	164	Standard query response 0x0676 AAAA fonts.googleapis.com CNAME googledadspl.l.google.com AAAA 2a00:1450:401b:804:200a
34239	903.643750	2001:730:3ed2:1000...	2a02:a317:e13b:8880...	DNS	149	Standard query response 0x4cd1 A fonts.gstatic.com CNAME gstaticadsl.l.google.com A 172.217.20.163
34244	903.645415	2a02:a317:e13b:8880...	2001:730:3ed2:153	DNS	94	Standard query 0x738f AAAA amalinowska.pl
34245	903.646197	2001:730:3ed2:1000...	2a02:a317:e13b:8880...	DNS	146	Standard query response 0x738f AAAA amalinowska.pl SOA ns1.nazwa.pl
34253	903.678474	2a02:a317:e13b:8880...	2001:730:3ed2:1000...	DNS	104	Standard query 0xe257 A ratings-wrs.symantec.com
34254	903.679188	2a02:a317:e13b:8880...	2001:730:3ed2:1000...	DNS	104	Standard query 0x4d26 AAAA ratings-wrs.symantec.com
34255	903.679310	2001:730:3ed2:153	2a02:a317:e13b:8880...	DNS	146	Standard query response 0x738f AAAA amalinowska.pl SOA ns1.nazwa.pl
34270	903.709731	2a02:a317:e13b:8880...	2001:730:3ed2:153	DNS	104	Standard query 0x4d26 AAAA ratings-wrs.symantec.com
34271	903.709788	2a02:a317:e13b:8880...	2001:730:3ed2:153	DNS	104	Standard query 0xe257 A ratings-wrs.symantec.com
34293	903.744888	2001:730:3ed2:1000...	2a02:a317:e13b:8880...	DNS	239	Standard query response 0x4d26 AAAA ratings-wrs.symantec.com CNAME shasta-wrs-prod.trafficmanager.net CNAME muel-shasta-wrs-pr
34298	903.760491	2001:730:3ed2:153	2a02:a317:e13b:8880...	DNS	239	Standard query response 0x4d26 AAAA ratings-wrs.symantec.com CNAME shasta-wrs-prod.trafficmanager.net CNAME muel-shasta-wrs-pr
34300	903.793858	2001:730:3ed2:153	2a02:a317:e13b:8880...	DNS	227	Standard query response 0xe257 A ratings-wrs.symantec.com CNAME shasta-wrs-prod.trafficmanager.net CNAME muel-shasta-wrs-prod
34303	903.836088	2001:730:3ed2:1000...	2a02:a317:e13b:8880...	DNS	227	Standard query response 0xe257 A ratings-wrs.symantec.com CNAME shasta-wrs-prod.trafficmanager.net CNAME muel-shasta-wrs-prod

> Frame 34303: 227 bytes on wire (1816 bits), 227 bytes captured (1816 bits) on interface 0

> Ethernet II, Src: CompalB8\_815a:c6 (ac:22:05:81:5a:c6), Dst: LiteOnTe\_ff:be:3f (98:22:ef:ff:be:3f)

> Internet Protocol Version 6, Src: 2001:730:3ed2:1000:153, Dst: 2a02:a317:e13b:8880:7d98:ef3a:ef76:7162

> User Datagram Protocol, Src Port: 53, Dst Port: 55333

> Domain Name System (response)

- Transaction ID: 0xe257
- Flags: 0x8180 Standard query response, No error
- Questions: 1
- Answer RRs: 3
- Authority RRs: 0
- Additional RRs: 0

> Queries

> Answers

[Request In: 34253]

[Time: 0.157614900 seconds]

### 3. Wnioski

Programy ping, traceroute i WireShark są bardzo przydatne do badania i analizowania sieci komputerowych. Dzięki nim zdecydowanie łatwiej wykrywać problemy na łączach, a co się z tym wiąże – rozwiązywać je. Program ping bardzo szybko pozwala nam stwierdzić czy problemy z siecią występują po stronie naszego komputera, czy serwera. Traceroute umożliwia nam zlokalizowanie problemu np. poprzez lokalizację wadliwego serwera. WireShark natomiast to bardzo dobre narzędzie do wyszukiwania błędów w danych przesyłanych w sieci. Każdy z tych programów przedstawia wyraźne korzyści, aby z nich korzystać.