Sprawozdanie
z listy zadań nr 1
z Technologii Sieciowych

Prowadzący: Dr Przemysław Kubiak

Konrad Grochowski

244936

1. Cel listy zadań

1.1 Przetestowanie programu Ping oraz wykonanie zadań dotyczących liczby węzłów na trasie do serwera (i z powrotem), wpływu wielkości pakietu oraz konieczności jego fragmentacji dla różnych odległości geograficznych od serwera;
1.2 Przetestowanie programu Traceroute oraz opis jego działania;
1.3 Przetestowanie programu WireShark oraz opis jego działania.

2. Opis programu Ping

Jest to oprogramowanie służące do testowania dostępności hosta w sieci IP, używane głównie do administracji sieci komputerowych.
Używając parametrów dla programu możemy określić m. in.:
- Liczbę wysyłanych zapytań do hosta (wraz z nieokreśloną, aż do przerwania);
- Wielkość wysyłanego pakietu;
- Dopuszczalny czas oczekiwania na pojedynczą odpowiedź;
- Wymuszenie odpowiedniego protokołu (IPv4 lub IPv6).
-TTL (ang. Time To Live) – dopuszczalną liczbę węzłów, przez którą pakiet może podróżować.

Do testów wybrałem dwa serwery:

- Serwer DNS o adresie 202.129.231.250 znajdujący się fizycznie na Fidżi
- Serwer serwisu onet.pl 213.180.141.140 znajdujący się fizycznie w Warszawie

2.1 Liczba węzłów na trasie

Liczbę węzłów na trasie do serwera, możemy uzyskać dzięki ustawieniu odpowiedniej wartości TTL: wartość X jest liczbą węzłów jeśli dla takiej wartości TTL pakiet nie dociera do hosta, lecz nie dociera przy wartości x-1.

Liczbę węzłów, przez które pakiet podróżuje z powrotem, możemy poznać przez odczyt wartości TTL pakietu zwrotnego: jest to domyślna wartość początkowa TTL serwera pomniejszona o rzeczoną liczbę węzłów.

Z poniższych zrzutów ekranu możemy wywnioskować, że pakiet wysłany do serwera na Fidżi przechodzi przez 21 węzłów, wracając przechodzi zaś przez 17, przyjmując domyślną wartość TTL serwera równą 64.

```
PS C:\WINDOWS\system32> ping 202.129.231.250 -i 20 -n 1

Pinging 202.129.231.250 with 32 bytes of data:
Reply from 45.117.244.170: TTL expired in transit.

Ping statistics for 202.129.231.250:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
PS C:\WINDOWS\system32>
```

```
PS C:\WINDOWS\system32> ping 202.129.231.250 -i 21 -n 1

Pinging 202.129.231.250 with 32 bytes of data:
Reply from 202.129.231.250: bytes=32 time=423ms TTL=47

Ping statistics for 202.129.231.250:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 423ms, Maximum = 423ms, Average = 423ms
```

Analogicznie, pakiet wysłany do serwera w Warszawie przechodzi przez 7 węzłów do serwera, a z powrotem przez 6, przyjmując tę samą wartość TTL serwera.

```
PS C:\WINDOWS\system32> ping onet.pl -i 6 -n 1

Pinging onet.pl [213.180.141.140] with 32 bytes of data:
Reply from 213.180.151.25: TTL expired in transit.
```

```
PS C:\WINDOWS\system32> ping onet.pl -i 7 -n 1

Pinging onet.pl [213.180.141.140] with 32 bytes of data:
Reply from 213.180.141.140: bytes=32 time=32ms TTL=58
```

2.2 Maksymalna wielkość niefragmentowanego pakietu

Maksymalną wielkość niefragmentowanego pakietu możemy zmierzyć poprzez ustawienie parametru wielkości pakietu wraz z wymuszeniem braku fragmentacji dzięki fladze „-f”. Zwrócone wyniki nie uwzględniają rozmiaru nagłówku ramki, musimy zatem dodać do nich 28 bajtów.

Fragmentacja zachodzi dla pakietów przekraczających MTU (ang. Maximum Transmission Unit), czyli maksymalnej dopuszczalnej wielkości pakietu. Zwykle wartość MTU oscyluje w granicach 1200-1500.

Testy na serwerze zlokalizowanym na Fidżi wykazały, maksymalną wielkość pakietu równą 1308 bajtów po dodaniu rzeczonej ramki.

Po podaniu większej wartości serwer zwraca komunikat o przekroczonej dopuszczalnej wartości niefragmentowanego pakietu.

```
PS C:\WINDOWS\system32> ping 202.129.231.250 -l 1280 -n 1

Pinging 202.129.231.250 with 1280 bytes of data:
Reply from 202.129.231.250: bytes=1280 time=407ms TTL=47

Ping statistics for 202.129.231.250:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 407ms, Maximum = 407ms, Average = 407ms
PS C:\WINDOWS\system32> ping 202.129.231.250 -l 1281 -n 1

Pinging 202.129.231.250 with 1281 bytes of data:
Reply from 202.129.231.250: Packet needs to be fragmented but DF set.

Ping statistics for 202.129.231.250:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
PS C:\WINDOWS\system32>
```

Test na serwerze zlokalizowanym w Warszawie zwrócił wynik 1460 bajtów. W przeciwieństwie do poprzedniego serwera, serwer nie zwraca komunikatu o przekroczeniu wartości, lecz nie odpowiada na takowy pakiet.

```
PS C:\WINDOWS\system32> ping onet.pl -l 1432  -f -n 1

Pinging onet.pl [213.180.141.140] with 1432 bytes of data:
Reply from 213.180.141.140: bytes=1432 time=28ms TTL=58

Ping statistics for 213.180.141.140:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 28ms, Maximum = 28ms, Average = 28ms
PS C:\WINDOWS\system32> ping onet.pl -l 1433  -f -n 1

Pinging onet.pl [213.180.141.140] with 1433 bytes of data:
Packet needs to be fragmented but DF set.

Ping statistics for 213.180.141.140:
    Packets: Sent = 1, Received = 0, Lost = 1 (100% loss),
PS C:\WINDOWS\system32>
```

```
PS C:\WINDOWS\system32> ping onet.pl -l 1600 -n 1

Pinging onet.pl [213.180.141.140] with 1600 bytes of data:
Request timed out.
```

2.2 Wpływ wielkości pakietów na czas przesyłania.

Z wywołań dla obu serwerów wynika, że wielkości niefragmentowanych pakietów nie wpływają na średni czas przesyłania.

```
PS C:\WINDOWS\system32> ping 202.129.231.250 -l 1200 -n 60

Ping statistics for 202.129.231.250:
    Packets: Sent = 60, Received = 60, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 405ms, Maximum = 488ms, Average = 416ms
```

```
PS C:\WINDOWS\system32> ping 202.129.231.250 -l 10 -n 60

Ping statistics for 202.129.231.250:
    Packets: Sent = 60, Received = 60, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 403ms, Maximum = 491ms, Average = 416ms
```

```
PS C:\WINDOWS\system32> ping onet.pl -l 10 -n 60

Ping statistics for 213.180.141.140:
    Packets: Sent = 60, Received = 60, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 14ms, Maximum = 90ms, Average = 21ms
```

```
PS C:\WINDOWS\system32> ping onet.pl -l 1000 -n 60

Ping statistics for 213.180.141.140:
    Packets: Sent = 60, Received = 60, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 15ms, Maximum = 45ms, Average = 20ms
```

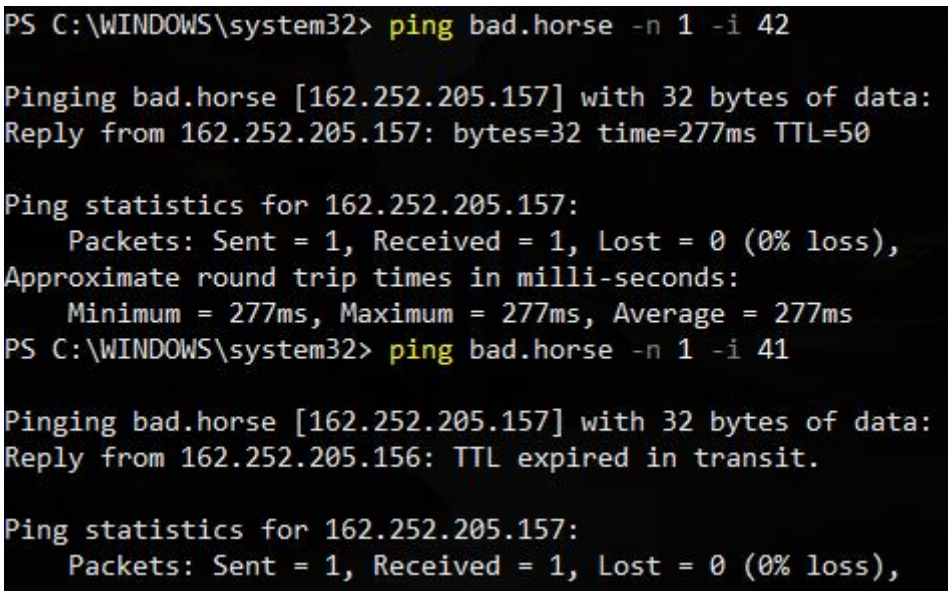
Oba serwery nie wysyłają oczekiwanej odpowiedzi przy wielkości pakietu przekraczającej MTU.

Odnalezienie serwera zwracającego pofragmentowane pakiety okazało się problematyczne. Wynika to z konfiguracji znaczącej wielkości serwerów, w skutek której protokół ICMP, który zawiera fragmentowane pakiety, zostaje ignorowany dla zachowania stabilności i bezpieczeństwa serwera.
Fragmentowane pakiety mogą mieć wpływ na stabilność przesyłania danych z uwagi na możliwość obrania przez nie różnych tras do serwera, przez co mogą zostać dostarczone w różnej kolejności.

2.4 „Średnica” internetu, liczba węzłów w sieciach wirtualnych.

Największa liczba węzłów, czyli tzw. średnica internetu, w rzeczywistości oscyluje w wartościach podobnych do serwera na Fidżi, tj. 21. Pakiety zwracające większą liczbę węzłów z dużym prawdopodobieństwem przechodzą przez sieci wirtualne.

Przykładowym serwerem opierającym się na połączeniu przez sieć wirtualną jest serwer na domenie „bad.horse”. Pakiet do tego serwera przechodzi przez 42 węzły.



2.5 Wnioski z analizy programu

Program Ping, mimo prostego interfejsu, okazuje się być niezbędnym oprogramowaniem do diagnostyki sieci, dzięki któremu możemy określić stabilność i prędkość infrastruktury pośredniczącej w komunikacji sieciowej oraz sposób, w jaki odpowiada na różną wielkość i strukturę przesyłanych pakietów.

3. Opis programu Traceroute

Jest to program służący do namierzenia adresów pośredniczących w przesyłaniu pakietu do serwera. Jego działanie polega na wysyłaniu pakietów o różnej wartości TTL do podanego serwera – serwery pośredniczące zwracają wtedy komunikat o zbyt niskiej

wartości TTL, podając również swój adres. Program domyślnie wysyła trzy sygnały do każdego węzła oraz zwraca każdy czas podróży w obie strony.

W programie możemy ustawić m. in. maksymalną liczbę węzłów, przez które może przejść pakiet, czas oczekiwania na wiadomość zwrotną oraz wymusić konkretny protokół.

3.1 Wywołania i analiza wyników

Poniższe wywołania zwracają adresy serwerów pośredniczących w przesyłaniu pakietu do testowanych wcześniej serwerów. Zwrócone wyniki potwierdzają oszacowane liczby węzłów.

```
PS C:\WINDOWS\system32> tracert  onet.pl

Tracing route to onet.pl [213.180.141.140]
over a maximum of 30 hops:

  1     1 ms     1 ms     1 ms  192.168.0.1
  2     *        *        *     Request timed out.
  3    17 ms    18 ms    23 ms  pl-ktw01a-rc1-ae18-0.aorta.net [84.116.253.129]
  4    17 ms    16 ms    16 ms  pl-krk07a-ra1-ae7-1400.aorta.net [84.116.193.25]
  5    22 ms    17 ms    19 ms  pni-pl-krk01a-as12990-onet.aorta.net [62.179.3.254]
  6    28 ms    41 ms    98 ms  sdr1.cdn1r1.z.j.ruc-br1.link3.net.onet.pl [213.180.151.25]
  7    16 ms    22 ms    15 ms  sg1.any.onet.pl [213.180.141.140]

Trace complete.
```

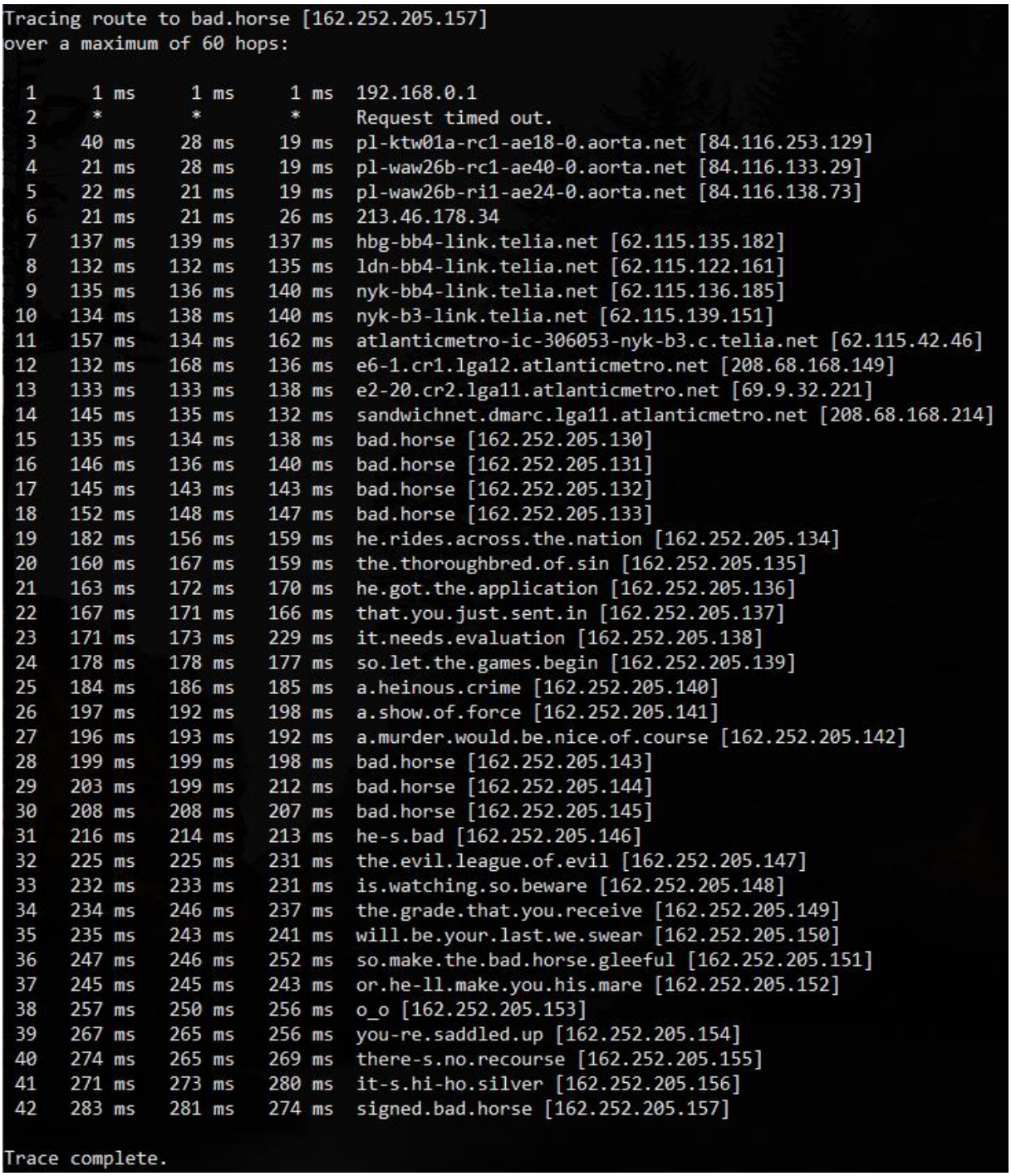
```
PS C:\WINDOWS\system32> tracert 202.129.231.250

Tracing route to 202.129.231.250 over a maximum of 30 hops

  1     2 ms     1 ms     1 ms  192.168.0.1
  2     *        *        *     Request timed out.
  3    18 ms    26 ms    23 ms  pl-ktw01a-rc1-ae18-0.aorta.net [84.116.253.129]
  4    21 ms    25 ms    20 ms  pl-waw26b-rc1-ae40-0.aorta.net [84.116.133.29]
  5    20 ms    22 ms    19 ms  pl-waw26b-ri1-ae24-0.aorta.net [84.116.138.73]
  6    22 ms    24 ms    20 ms  ae-13.r01.wrswpl01.pl.bb.gin.ntt.net [129.250.9.109]
  7    44 ms    44 ms    40 ms  ae-12.r24.amstnl02.nl.bb.gin.ntt.net [129.250.3.81]
  8    44 ms    44 ms    40 ms  ae-3.r25.amstnl02.nl.bb.gin.ntt.net [129.250.4.69]
  9   141 ms   144 ms   147 ms  ae-5.r23.asbnva02.us.bb.gin.ntt.net [129.250.6.162]
 10   206 ms   211 ms   219 ms  ae-10.r22.snjsca04.us.bb.gin.ntt.net [129.250.6.237]
 11   206 ms   214 ms   207 ms  ae-40.r02.snjsca04.us.bb.gin.ntt.net [129.250.3.121]
 12   207 ms   204 ms   208 ms  ae-4.r06.plalca01.us.bb.gin.ntt.net [129.250.4.118]
 13   211 ms   210 ms   208 ms  ae-0.tnzi.plalca01.us.bb.gin.ntt.net [129.250.203.42]
 14   209 ms   317 ms   238 ms  ae0-3.sjbr3.global-gateway.net.nz [203.96.120.73]
 15   366 ms   368 ms   363 ms  122.56.127.30
 16   363 ms   368 ms   370 ms  ae2-10.sgbr4.global-gateway.net.nz [202.50.232.246]
 17   373 ms   368 ms   374 ms  skytv-int-sec.tkbr4.global-gateway.net.nz [202.50.238.62]
 18     *        *        *     Request timed out.
 19   407 ms   406 ms   400 ms  45.117.244.169
 20   405 ms   405 ms   401 ms  45.117.244.170
 21   416 ms   419 ms   408 ms  202.129.231.250

Trace complete.
```

Wykorzystanie programu w celu zbadania adresu bad.horse zwraca domeny tworzące tekst piosenki oraz zbliżone adresy IPv4, co może sugerować, iż część węzłów składa się na sieć wirtualną.



```
Tracing route to bad.horse [162.252.205.157]
over a maximum of 60 hops:

  1     1 ms     1 ms     1 ms  192.168.0.1
  2     *        *        *     Request timed out.
  3    40 ms    28 ms    19 ms  pl-ktw01a-rc1-ae18-0.aorta.net [84.116.253.129]
  4    21 ms    28 ms    19 ms  pl-waw26b-rc1-ae40-0.aorta.net [84.116.133.29]
  5    22 ms    21 ms    19 ms  pl-waw26b-ri1-ae24-0.aorta.net [84.116.138.73]
  6    21 ms    21 ms    26 ms  213.46.178.34
  7   137 ms   139 ms   137 ms  hbg-bb4-link.telia.net [62.115.135.182]
  8   132 ms   132 ms   135 ms  ldn-bb4-link.telia.net [62.115.122.161]
  9   135 ms   136 ms   140 ms  nyk-bb4-link.telia.net [62.115.136.185]
 10   134 ms   138 ms   140 ms  nyk-b3-link.telia.net [62.115.139.151]
 11   157 ms   134 ms   162 ms  atlanticmetro-ic-306053-nyk-b3.c.telia.net [62.115.42.46]
 12   132 ms   168 ms   136 ms  e6-1.cr1.lga12.atlanticmetro.net [208.68.168.149]
 13   133 ms   133 ms   138 ms  e2-20.cr2.lga11.atlanticmetro.net [69.9.32.221]
 14   145 ms   135 ms   132 ms  sandwichnet.dmarc.lga11.atlanticmetro.net [208.68.168.214]
 15   135 ms   134 ms   138 ms  bad.horse [162.252.205.130]
 16   146 ms   136 ms   140 ms  bad.horse [162.252.205.131]
 17   145 ms   143 ms   143 ms  bad.horse [162.252.205.132]
 18   152 ms   148 ms   147 ms  bad.horse [162.252.205.133]
 19   182 ms   156 ms   159 ms  he.rides.across.the.nation [162.252.205.134]
 20   160 ms   167 ms   159 ms  the.thoroughbred.of.sin [162.252.205.135]
 21   163 ms   172 ms   170 ms  he.got.the.application [162.252.205.136]
 22   167 ms   171 ms   166 ms  that.you.just.sent.in [162.252.205.137]
 23   171 ms   173 ms   229 ms  it.needs.evaluation [162.252.205.138]
 24   178 ms   178 ms   177 ms  so.let.the.games.begin [162.252.205.139]
 25   184 ms   186 ms   185 ms  a.heinous.crime [162.252.205.140]
 26   197 ms   192 ms   198 ms  a.show.of.force [162.252.205.141]
 27   196 ms   193 ms   192 ms  a.murder.would.be.nice.of.course [162.252.205.142]
 28   199 ms   199 ms   198 ms  bad.horse [162.252.205.143]
 29   203 ms   199 ms   212 ms  bad.horse [162.252.205.144]
 30   208 ms   208 ms   207 ms  bad.horse [162.252.205.145]
 31   216 ms   214 ms   213 ms  he-s.bad [162.252.205.146]
 32   225 ms   225 ms   231 ms  the.evil.league.of.evil [162.252.205.147]
 33   232 ms   233 ms   231 ms  is.watching.so.beware [162.252.205.148]
 34   234 ms   246 ms   237 ms  the.grade.that.you.receive [162.252.205.149]
 35   235 ms   243 ms   241 ms  will.be.your.last.we.swear [162.252.205.150]
 36   247 ms   246 ms   252 ms  so.make.the.bad.horse.gleeful [162.252.205.151]
 37   245 ms   245 ms   243 ms  or.he-ll.make.you.his.mare [162.252.205.152]
 38   257 ms   250 ms   256 ms  o_o [162.252.205.153]
 39   267 ms   265 ms   256 ms  you-re.saddled.up [162.252.205.154]
 40   274 ms   265 ms   269 ms  there-s.no.recourse [162.252.205.155]
 41   271 ms   273 ms   280 ms  it-s.hi-ho.silver [162.252.205.156]
 42   283 ms   281 ms   274 ms  signed.bad.horse [162.252.205.157]

Trace complete.
```

3.2 Wnioski z analizy programu

Program Traceroute w paru funkcjonalnościach pokrywa się z programem Ping. Umożliwia on jednak sprawne wyszukanie adresów pośredniczących węzłów oraz znalezienie wąskiego gardła w infrastrukturze, zachowując równie prosty interfejs.

4. Opis programu WireShark

Jest to oprogramowanie służący do przechwytywania, nagrywania i dekodowania pakietów przechodzących przez całą sieć lokalną. Program zapisuje w postaci listy wszystkie pakiety, które przechwytuje. Mamy możliwość zapisania i odczytywania różnych sesji przechwytywania. Jest też wyposażony w zaawansowane narzędzia filtracji pakietów; możemy np. wyszukiwać konkretne dialogi po protokołach TCP, UDP, TLS i http.

4.1 Przypadki użycia

4.1.1 Wycinek z długiej sesji przechwytywania na sieci lokalnej, gdzie adresatami pakietów są również inne urządzenia:

567127	18828.981004	192.168.0.74	239.255.255.250	SSDP	215 M-SEARCH * HTTP/1.1
567128	18829.296150	IntelCor_d1:97:5a	Broadcast	ARP	60 Who has 192.168.0.1? Tell 192.168.0.59
567129	18829.300444	IntelCor_d1:97:5a	Broadcast	ARP	60 Who has 192.168.0.1? Tell 192.168.0.59
567130	18829.311386	IntelCor_d1:97:5a	Broadcast	ARP	60 Who has 192.168.0.1? Tell 192.168.0.59
567131	18829.982175	192.168.0.74	239.255.255.250	SSDP	215 M-SEARCH * HTTP/1.1
567132	18830.768502	2a02:a317:e141:7900…	2a03:2880:f016:b:fa…	TLSv1.3	106 Application Data
567133	18830.805826	2a03:2880:f016:b:fa…	2a02:a317:e141:7900…	TCP	74 443 → 2023 [ACK] Seq=13978 Ack=10854 Win=34560 Len=0
567134	18830.843267	2a03:2880:f016:b:fa…	2a02:a317:e141:7900…	TLSv1.3	102 Application Data
567135	18830.883001	2a02:a317:e141:7900…	2a03:2880:f016:b:fa…	TCP	74 2023 → 443 [ACK] Seq=10854 Ack=14006 Win=260864 Len=0
567136	18830.990092	192.168.0.74	239.255.255.250	SSDP	215 M-SEARCH * HTTP/1.1
567137	18831.555920	2a01:111:2010:6::ff…	2a02:a317:e141:7900…	TCP	74 443 → 13099 [RST, ACK] Seq=23438 Ack=8022 Win=0 Len=0
567138	18831.801824	2a02:a317:e141:7900…	2a03:2880:f016:b:fa…	TLSv1.3	106 Application Data
567139	18831.821526	2a03:2880:f016:b:fa…	2a02:a317:e141:7900…	TCP	74 443 → 2021 [ACK] Seq=13922 Ack=13200 Win=45568 Len=0
567140	18831.872232	2a03:2880:f016:b:fa…	2a02:a317:e141:7900…	TLSv1.3	102 Application Data
567141	18831.911622	2a02:a317:e141:7900…	2a03:2880:f016:b:fa…	TCP	74 2021 → 443 [ACK] Seq=13200 Ack=13950 Win=261120 Len=0

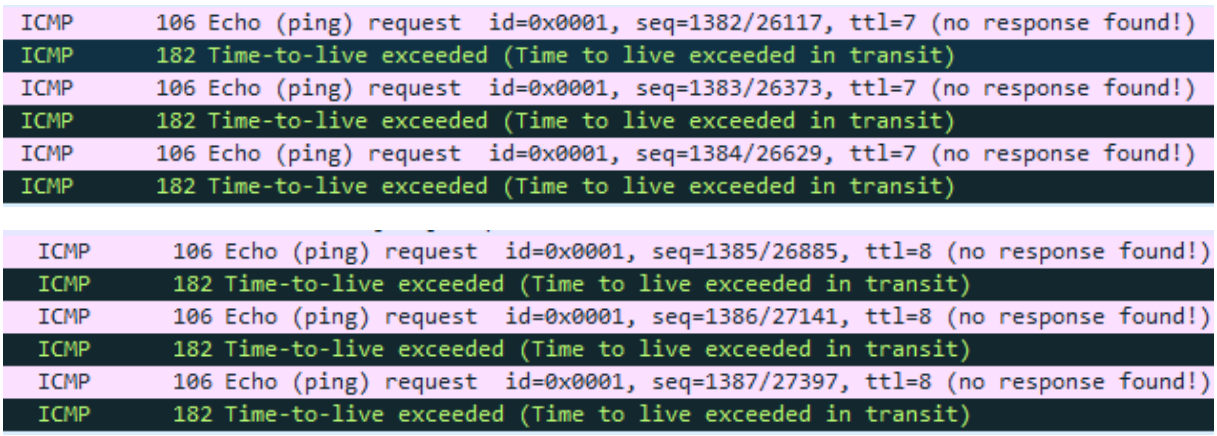
4.1.2 Wymiana pakietów między urządzeniami w protokole TCP:

759857	19824.525391	192.168.0.39	104.74.103.204	TCP	66 13288 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
759858	19824.554887	104.74.103.204	192.168.0.39	TCP	66 80 → 13288 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1420 SACK_PERM=1 WS=128
759859	19824.554994	192.168.0.39	104.74.103.204	TCP	54 13288 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0
759860	19824.555100	192.168.0.39	104.74.103.204	HTTP	267 GET /pl-PL/livetile/preinstall?region=PL&appid=C98EA5B0842DBB9405BBF071E1DA76512D21FE36&FORM=Threshold HTTP/1.1
759861	19824.588313	104.74.103.204	192.168.0.39	TCP	60 80 → 13288 [ACK] Seq=1 Ack=214 Win=30336 Len=0
759862	19824.590414	104.74.103.204	192.168.0.39	TCP	1474 80 → 13288 [ACK] Seq=1 Ack=214 Win=30336 Len=1420 [TCP segment of a reassembled PDU]
759863	19824.590834	104.74.103.204	192.168.0.39	TCP	1474 80 → 13288 [ACK] Seq=1421 Ack=214 Win=30336 Len=1420 [TCP segment of a reassembled PDU]
759864	19824.590862	192.168.0.39	104.74.103.204	TCP	54 13288 → 80 [ACK] Seq=214 Ack=2841 Win=262656 Len=0
759865	19824.591234	104.74.103.204	192.168.0.39	TCP	1474 80 → 13288 [ACK] Seq=2841 Ack=214 Win=30336 Len=1420 [TCP segment of a reassembled PDU]
759866	19824.591648	104.74.103.204	192.168.0.39	HTTP/X…	398 HTTP/1.1 200 OK
759867	19824.591675	192.168.0.39	104.74.103.204	TCP	54 13288 → 80 [ACK] Seq=214 Ack=4605 Win=262656 Len=0
762067	19884.592200	192.168.0.39	104.74.103.204	TCP	54 13288 → 80 [FIN, ACK] Seq=214 Ack=4605 Win=262656 Len=0
762074	19884.624280	104.74.103.204	192.168.0.39	TCP	60 80 → 13288 [FIN, ACK] Seq=4605 Ack=215 Win=30336 Len=0
762075	19884.624336	192.168.0.39	104.74.103.204	TCP	54 13288 → 80 [ACK] Seq=215 Ack=4606 Win=262656 Len=0

4.1.3 Przechwytywanie pakietów z programu Ping po protokole ICMP wykorzystywanym do diagnostyki sieci:

763663	20036.557129	192.168.0.39	213.180.141.140	ICMP	74 Echo (ping) request id=0x0001, seq=1322/10757, ttl=128 (reply in 763664)
763664	20036.573246	213.180.141.140	192.168.0.39	ICMP	74 Echo (ping) reply id=0x0001, seq=1322/10757, ttl=58 (request in 763663)
763665	20037.563340	192.168.0.39	213.180.141.140	ICMP	74 Echo (ping) request id=0x0001, seq=1323/11013, ttl=128 (reply in 763666)
763666	20037.583494	213.180.141.140	192.168.0.39	ICMP	74 Echo (ping) reply id=0x0001, seq=1323/11013, ttl=58 (request in 763665)

4.1.4 Przechwytywanie pakietów z programu Traceroute (wartość TTL inkrementowana dla zidentyfikowania każdego węzła):



4.2 Wnioski z analizy programu

Program WireShark jest nieocenionym narzędziem przy kontrolowaniu ruchu w sieci lokalnej. Kluczowym atutem jest analizowanie wymiany pakietów między poszczególnymi dwoma hostami. Dopełnia on funkcjonalność poprzednich programów poprzez możliwość badania zawartości wysyłanych przez nie pakietów.
Domyślnie dostarcza informacji o całym ruchu w sieci, więc wyszukanie odpowiednich pakietów do analizy może okazać się problematyczne, pomimo bogatych narzędzi do filtracji.