Sprawozdanie  
z listy zadań nr 1  
z Technologii Sieciowych

Prowadzący: Dr Przemysław Kubiak

Konrad Grochowski

244936

1. **Cel listy zadań**
   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;
   2. Przetestowanie programu Traceroute oraz opis jego działania;
   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

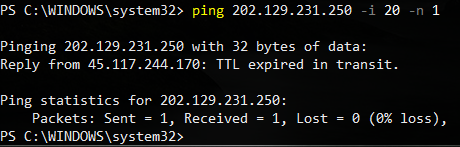
- Serwer serwisu onet.pl 213.180.141.140 znajdujący się fizycznie w Warszawie

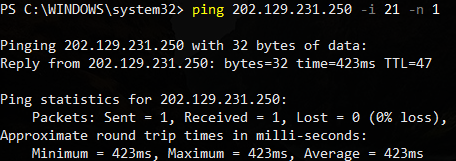
* 1. **Liczba węzłów na trasie**

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.

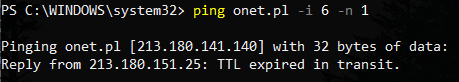
Liczbę węzłów, przez które pakiet podróżuje z powrotem, możemy poznać przez odczyt wartości TTL pakietu zwrotnego: jest to domyślna wartość początkowa TTL serwera pomniejszona o rzeczoną liczbę węzłów.

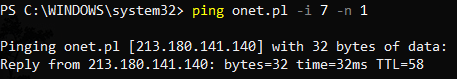
Z poniższych zrzutów ekranu możemy wywnioskować, że pakiet wysłany do serwera na Fidżi przechodzi przez 21 węzłów, wracając przechodzi zaś przez 17, przyjmując domyślną wartość TTL serwera równą 64.





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



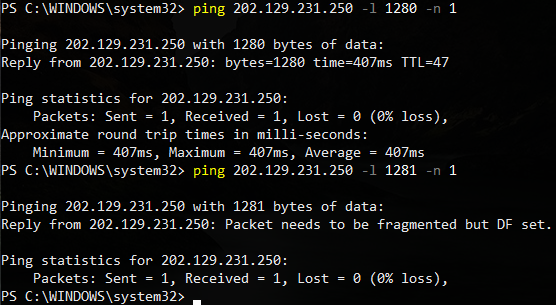


* 1. **Maksymalna wielkość niefragmentowanego pakietu**

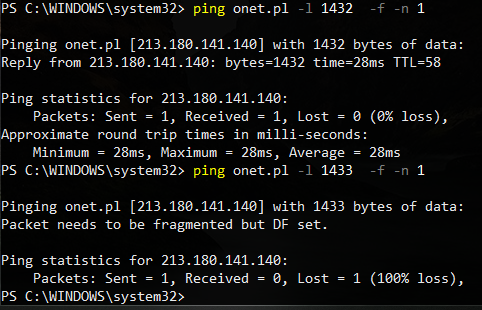
Maksymalną wielkość niefragmentowanego pakietu możemy zmierzyć poprzez ustawienie parametru wielkości pakietu wraz z wymuszeniem braku fragmentacji dzięki fladze „-f”. Zwrócone wyniki nie uwzględniają rozmiaru nagłówku ramki, musimy zatem dodać do nich 28 bajtów.  
Fragmentacja zachodzi dla pakietów przekraczających MTU (ang. Maximum Transmission Unit), czyli maksymalnej dopuszczalnej wielkości pakietu. Zwykle wartość MTU oscyluje  
w granicach 1200-1500.

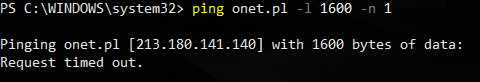
Testy na serwerze zlokalizowanym na Fidżi wykazały, maksymalną wielkość pakietu równą 1308 bajtów po dodaniu rzeczonej ramki.

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



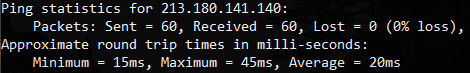
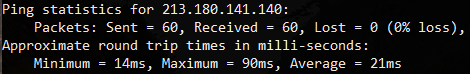
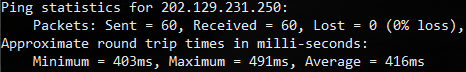
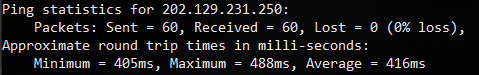
Test na serwerze zlokalizowanym w Warszawie zwrócił wynik 1460 bajtów.  
W przeciwieństwie do poprzedniego serwera, serwer nie zwraca komunikatu o przekroczeniu wartości, lecz nie odpowiada na takowy pakiet.





**2.2 Wpływ wielkości pakietów na czas przesyłania.**

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



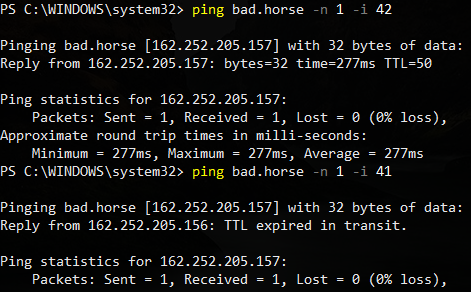
Oba serwery nie wysyłają oczekiwanej odpowiedzi przy wielkości pakietu przekraczającej MTU.

Nie udało mi się znaleźć serwera, który zwraca pofragmentowane pakiety. Wynika to  
z konfiguracji znaczącej wielkości serwerów, w skutek której protokół ICMP, który zawiera fragmentowane pakiety, zostaje ignorowany dla zachowania stabilności i bezpieczeństwa serwera.

**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.

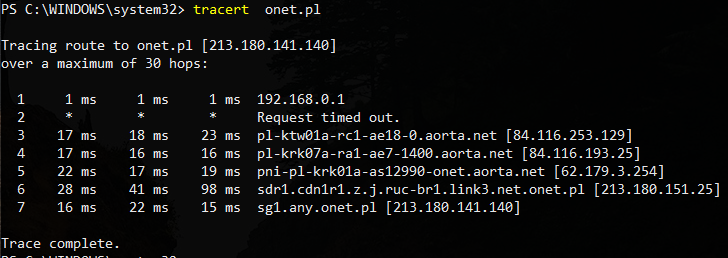
1. **Opis programu Traceroute**

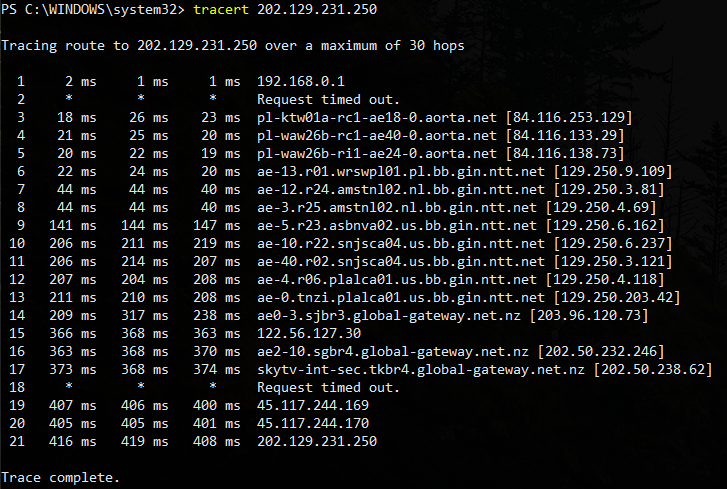
Jest to program służący do namierzenia adresów pośredniczących w przesyłaniu pakietu do serwera. Jego działanie polega na wysyłaniu pakietów o różnej wartości TTL  
do podanego serwera – serwery pośredniczące zwracają wtedy komunikat o zbyt niskiej wartości TTL, podając również swój adres. Program domyślnie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

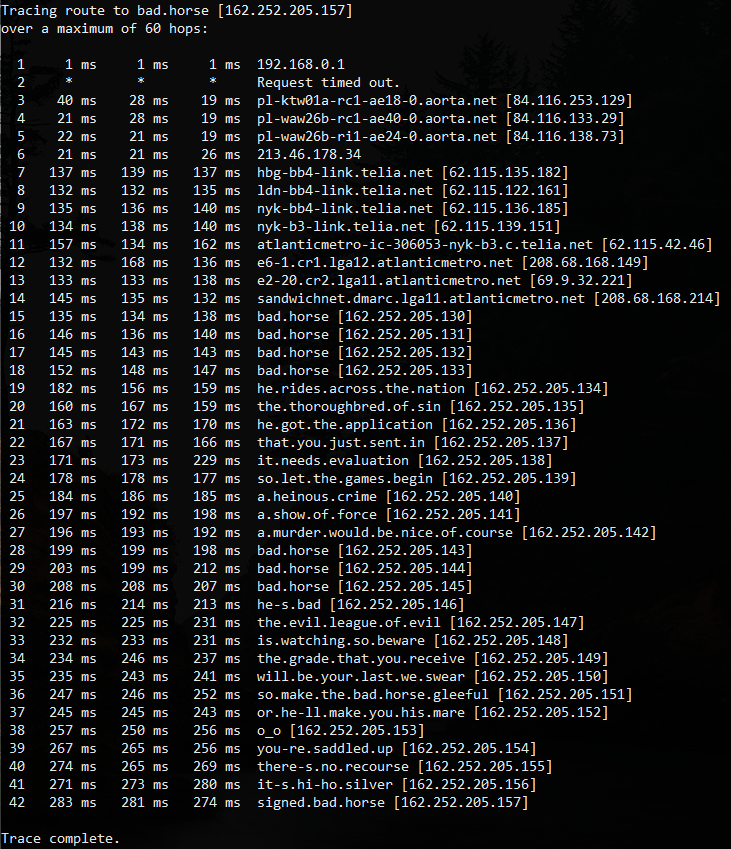
**3.1 Wywołania i analiza wyników**

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





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.



**3.2 Wnioski z analizy programu**

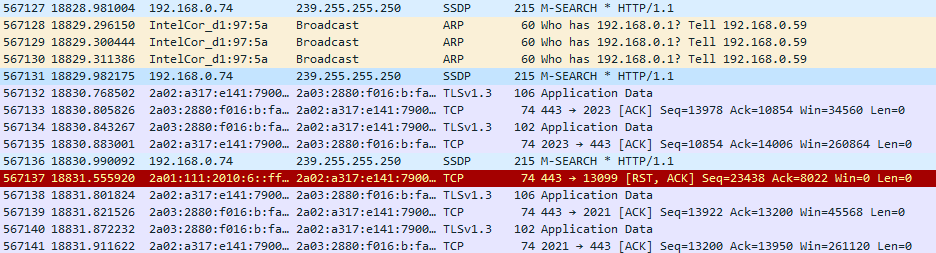
Program Traceroute w paru funkcjonalnościach pokrywa się z programem Ping. Umożliwia on jednak sprawne wyszukanie adresów pośredniczących węzłów oraz znalezienie wąskiego gardła w infrastrukturze, zachowując równie prosty interfejs.

1. **Opis programu WireShark**

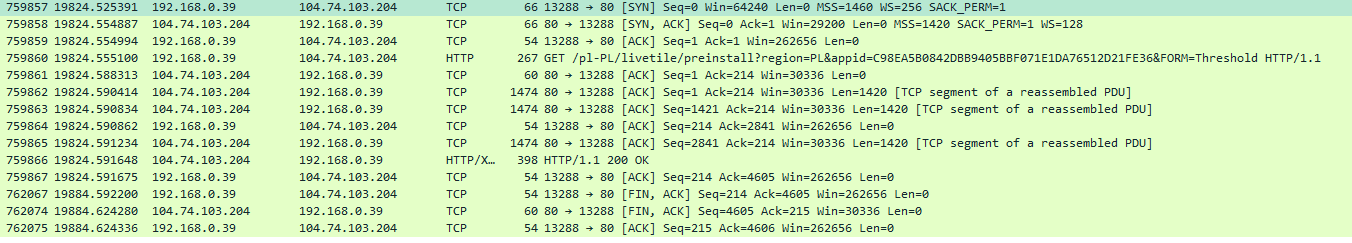
Jest to oprogramowanie służący do przechwytywania, nagrywania i dekodowania pakietów przechodzących przez całą sieć lokalną. Program zapisuje w postaci listy wszystkie pakiety, które przechwytuje. Mamy możliwość zapisania i odczytywania różnych sesji przechwytywania. Jest też wyposażony w zaawansowane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

**4.1 Przypadki użycia**

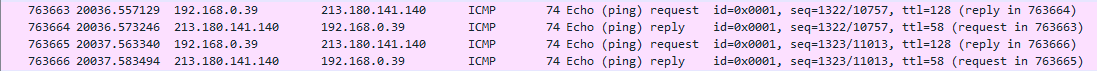
4.1.1 Wycinek z długiej sesji przechwytywania na sieci lokalnej, gdzie adresatami pakietów są również inne urządzenia:



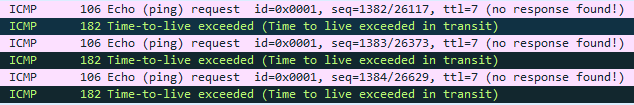
4.1.2 Wymiana pakietów między urządzeniami w protokole TCP:

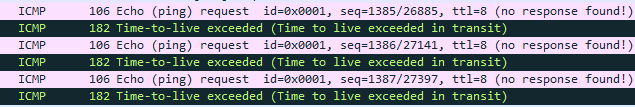


4.1.3 Przechwytywanie pakietów z programu Ping po protokole ICMP wykorzystywanym do diagnostyki sieci:



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.